ACCURACY-ENERGY TRADEOFFS IN DIGITAL IMAGE PROCESSING USING EMBEDDED COMPUTING PLATFORMS

A Thesis
Presented to
The Academic Faculty

by

Se Hun Kim

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2011

ACCURACY-ENERGY TRADEOFFS IN DIGITAL IMAGE PROCESSING USING EMBEDDED COMPUTING PLATFORMS

Approved by:

Dr. Marilyn Wolf, Advisor
School of Electrical and Computer Eng.
Georgia Institute of Technology

Dr. Patricio Vela
School of Electrical and Computer Eng.
Georgia Institute of Technology

Dr. David Anderson
School of Electrical and Computer Eng.
Georgia Institute of Technology

Dr. Hyesoon Kim
School of Computer Science
Georgia Institute of Technology

Dr. Saibal Mukhopadhyay
School of Electrical and Computer Eng.
Georgia Institute of Technology

Date Approved: November 1, 2011

To my parents

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CMOS	complementary metal–oxide–semiconductor
D2D	die-to-die
DCT	discrete cosine transform
DRAM	dynamic random-access memory
DSP	digital signal processing
EDA	electronic design automation
FA	fulladder
FSM	finite state machine
FDCT	full 2-D DCT
HOB	high-order bit
IC	integrated circuit
ITRS	international technology roadmap for semiconductor
JPEG	joint photographic experts group
LOB	low-order bit
LSB	least significant bit
MAC	multiplier–accumulator
MSB	most significant bit
MSE	mean squared error
MSSIM	mean structural similarity
PSNR	peak signal-to-noise ratio
PTM	predictive technology model
RCDCT	reduced coefficient 2-D DCT

SEFF	soft-edge flip-flop
SRAM	static random-access memory
WID	within-die

SUMMARY

As more and more multimedia applications are integrated in mobile devices, a significant amount of energy is devoted to digital signal processing (DSP). Thus, reducing energy consumption for DSP systems has become an important design goal for battery operated mobile devices. Since supply voltage scaling is one of the most effective methods to reduce power/energy consumption, this study examines aggressive voltage scaling to achieve significant energy savings by allowing some output quality degradation for error tolerant image processing system. The objective of proposed research is to explore ultra-low energy image processing system design methodologies based on efficient accuracy (quality)-energy tradeoffs.

This dissertation presents several new analyses and techniques to achieve significant energy savings without noticeable quality degradation under aggressive voltage scaling. In the first, this work starts from accurate error analysis and a model based on input sequence dependent delay estimation. Based on the analysis, we explain the dependence of voltage scalability on input image types, which may be used for input dependent adaptive control for optimal accuracy-energy tradeoffs. In addition, this work includes the system-level analysis of the impact of aggressive voltage scaling on overall energy consumption and a low-cost technique to reduce overall energy consumption. Lastly, this research exploits an error concealment technique to improve the efficiency of accuracy-energy tradeoffs. For an image compression system, the technique minimizes the impact of delay errors on output quality while allowing very low voltage operations for significant energy reduction.

CHAPTER 1

INTRODUCTION

Semiconductor technology has been tremendously improved ever since the first integrated circuit (IC) was introduced in the 1960's. IC design has been focused on high performance computing and small chip area. However, as portable device market increases exponentially, energy consumption has got attention more than ever before. Figure 1.1 shows the exponential growth in the number of global cellular phone subscribers, and Figure 1.2 shows the predicted sale of media tablet products. Energy consumption is directly related to the runtime of the battery operated devices as well as the weight and volume of the devices. International technology roadmap for semiconductor (ITRS) reported that the battery life has declined as more and more new features are added into the devices while the development of battery technology is much slower than the increase in the functional requirements [1]. For the reason, reducing energy consumption has become one of the most important design goals for the devices.

As more and more multimedia applications are integrated in the mobile devices, a significant amount of energy is devoted to digital signal processing (DSP). Numerous

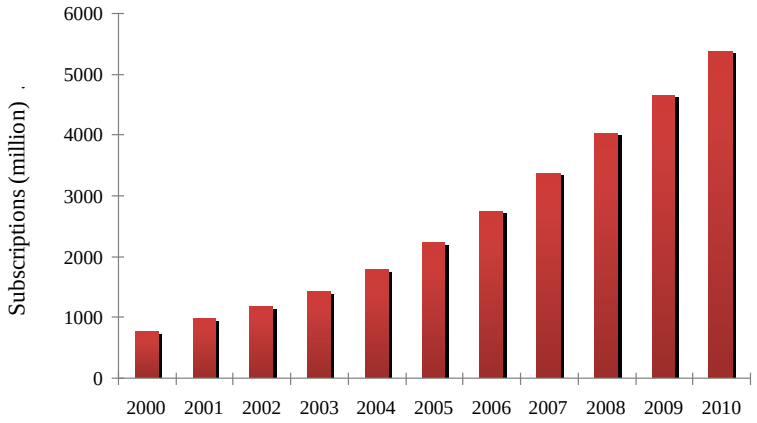


Figure 1.1: Global growth in the number of mobile cellular phone subscribers [2].

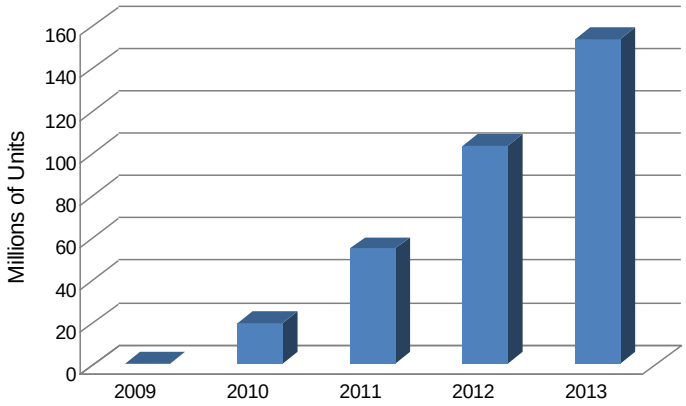


Figure 1.2: Global media tablet sale forecast [3].

techniques for low energy DSP systems have been sought at various levels of design such as algorithm, architecture, and circuit. For example, during the implementation of DSP algorithms, redundant computations are searched and eliminated. In addition, complex computations such as multiplication can be replaced by simple computations such as shift and addition [4]. Algorithmic transformations such as pipelining and parallel processing reduce power/energy consumption by utilizing lower supply voltages [5-7]. Since the choice of DSP primitives (i.e. adders and multipliers) affects energy consumption as well as performance of the system, the efficient implementation of these primitives on the circuit level also have been discussed in many previous studies [8].

The low energy techniques can be divided into two different categories; Ones that do not introduce errors and ones that do. Several conventional approaches explained above attempted to reduce energy consumption without any impact on output quality. In contrast, techniques based on the error tolerance of DSP applications explore energy savings by trading off the accuracy of computations. The output quality of the applications is often determined by the human perceptual system, which masks small errors [9]. This brings inherent error tolerance for such applications. The error tolerance property of the applications has been used by various previous approaches to optimize the

performance and the energy consumption of the system by allowing slight degradation in output quality. A simple example of these approaches includes the trade-off between the bit-width of computations and hardware requirement. The some extent of inaccurate computation may not result in noticeable output quality degradation while achieving low energy consumption from reduced hardware requirement. In this case, the magnitude of error is the difference between the reduced bit-width and normal computation. In addition, the relaxation of the requirement of 100% correctness in operation may significantly reduce energy consumption by allowing aggressive voltage scaling, which causes erroneous operations.

It is well known that supply voltage scaling is an effective technique for reducing the power/energy consumption of CMOS integrated circuits because it reduces switching, short-circuit, and leakage power [10]. Recently, aggressive voltage scaling techniques have been widely proposed to achieve significant reduction in energy consumption. However, operations at a voltage lower than the critical voltage ($V_{dd\text{-}crit}$), which ensures correct functionality, leads to timing failures because of increased delay [11][18]. Numerous previous studies presented error detection and correction/compensation techniques while applying aggressive voltage scaling for great energy savings [11-17]. These studies are mostly focused on error control techniques without an appropriate analysis of the characteristics of error and their impacts on output quality under aggressive voltage scaling.

The objective of this research is to provide ultra-low energy image processing design methodologies based on efficient accuracy-energy tradeoff under aggressive voltage scaling and process variation. The main contributions of this research are

i) A new error analysis and modeling using accurate delay estimation under aggressive voltage scaling,

ii) The exploration of the relationship between errors and image characteristics to improve the efficiency of accuracy-energy tradeoffs,

iii) The analysis of the implication of erroneous operations on overall energy consumption and energy optimization through system-level exploration,

iv) An adaptive error concealing method for efficient accuracy-energy tradeoffs for image compression.

To find an optimal accuracy-energy tradeoff point under aggressive voltage scaling, a fundamental step is the analysis and the modeling of the error characteristics based on the accurate delay estimation under aggressive voltage scaling. As pointed out in [8], the error rate depends on the frequency of the excitation of critical or subcritical paths. It is important to note that the excitations are dependent on not only combinational input but also the previous state of logic. Therefore, an accurate error estimation method should be based on the transition delay on the circuit level, which is related to the input sequence. Based on the analysis and the modeling, we explain an experimental framework that allows the accurate estimation of error occurrence and energy consumption within reasonable time.

Based on the input sequence dependent error analysis, we show that the error resilience of images depends on their content as well as the distribution of erroneous operations. The analysis of the relationship between the characteristics of input image types and quality degradation allows the adaptive control of parameters such as the voltage level and the bit-width dependent on the characteristics so as to realize efficient accuracy-energy tradeoffs under aggressive voltage scaling. In addition, this research exploits the system-level energy optimization by investigating the impact of errors on other subsystems. For example of image compression systems, errors in discrete cosine transform (DCT) under aggressive voltage scaling decrease compression ratio, which causes significant energy overhead for storing compressed image. To maximize overall energy savings in system level, we introduce a very simple but effective error reduction technique based on bit truncation. This technique allows significantly reduced overall quality degradation at the same voltage level.

For the last, to improve the efficiency of the tradeoffs further, the system-level exploration of error concealment is proposed for the image processing system. The proposed reconfigurable architecture allows aggressive voltage scaling to achieve great energy savings while minimizing the impact of the delay errors by concealing the errors. Based on the analysis of the characteristics of erroneous DCT output, a simple error detection method, which requires trivial implementation cost, is presented. Since the accuracy-energy tradeoff occurs only at the presence of an error, the proposed approach may maintain high image quality under very low voltages.

Thesis organization

The thesis is organized into seven chapters.

Chapter 1: INTRODUCTION. This chapter introduces energy consumption issue for digital signal processing hardware design. It also summarizes the contributions of this thesis.

Chapter 2: ORIGIN AND HISTORY OF THE PROBLEM. This chapter explains background materials and describes previous work in low energy design based on voltage scaling.

Chapter 3: ERROR ANALYSIS UNDER SCALED VOLTAGE. This chapter discusses the impact of voltage scaling on circuit behavior and an error analysis and model based on the accurate delay estimation under scaled voltages.

Chapter 4: PERCEPTION-BASED ERROR TOLERANCE AND ENERGY SAVINGS. This chapter explains the relationship between input image type and voltage scalability based on detailed experimental analysis. It discusses two main reasons for the relationship and its impact on energy savings.

Chapter 5: SYSTEM-LEVEL ENERGY ANALYSIS AND OPTIMIZATION. This chapter discusses the system-level analysis of aggressive voltage scaling on energy savings and a low cost error reduction technique.

Chapter 6: EFFICIENT ACCURACY-ENERGY TRADEOFF BASED ON ERROR CONCEALMENT. This chapter discusses efficient accuracy-energy tradeoffs with an adaptive error concealment technique that allows very low voltage operations while maintaining a high quality level.

Chapter 7: CONCLUSION. This chapter summarizes the major accomplishments of this dissertation and suggests future work.

CHAPTER 2

ORIGIN AND HISTORY OF THE PROBLEM

2.1 Introduction

This chapter provides disjoint background materials directly related to this work to render this document self-contained. First, since our work is focused on low energy design methodology, the sources of power/energy consumption are presented briefly. Then, we explain conventional low power/energy design techniques especially for DSP applications. In this dissertation, supply voltage scaling is mainly considered for effectively reducing energy consumption, so we discuss issues due to supply voltage scaling and introduce several previous studies that achieve energy savings using aggressive voltage scaling in the last section of this chapter. This chapter also presents general explanation about an image compression system and the mathematical definition of discrete cosine transform. It also includes the definitions of several image quality metrics such as conventional error sensitivity based metrics and a perceptual image quality metric.

2.2 Power/energy consumption

The total power consumption of digital CMOS circuits includes three main sources as shown in equation below: switching power consumption, short-circuit power consumption, and leakage power consumption.

$$P_{total} = P_{switching} + P_{short-circuit} + P_{leakage} \tag{2.1}$$

Switching power consumption is the power that is consumed in the charging and the discharging of capacitances through normal circuit operations. As shown in Figure 2.1 (a), during the falling transition at the input node of the inverter, the current (i_{cap}) flows

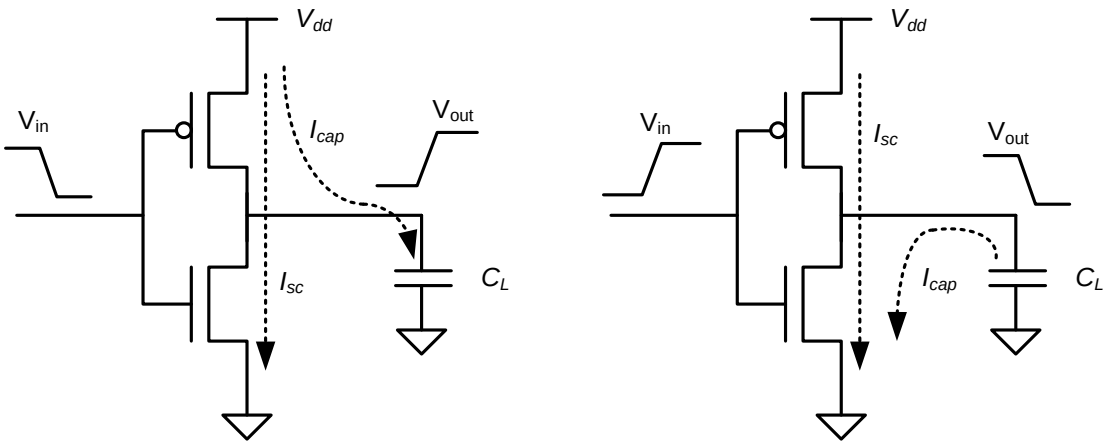


Figure 2.1: Illustration of the sources of dynamic power consumption.

through the PMOS from the power supply to the output node, and the capacitor (C_L) is charged. The total energy consumption is explained the equation below.

$$E = V_{dd}\int_{o}^{T} i_{cap}(t)dt = V_{dd}\int_{o}^{V_{dd}} C_L dV_{out} = C_L V_{out} \tag{2.2}$$

As shown in the Figure 2.1 (b), during the raising transition at the input node, the stored charge dissipated through NMOS. Thus, the average switching power consumption is given by

$$P_{switching} = C_L \cdot V_{dd}^2 \cdot f_{avg}, \tag{2.3}$$

where f_{avg} is the average frequency of zero-to-one transitions. This often expressed with the probability of the transition (switching activity factor: $\alpha_{0\to 1}$) in a clock period. Equation 2.3 can be rewritten as

$$P_{switching} = \alpha_{0\to 1} \cdot f_{clk} \cdot C_L \cdot V_{dd}^2 \tag{2.4}$$

In ideal digital CMOS circuits, pull-up and pull-down networks never conduct simultaneously. However, in reality, when the output of a gate changes and pull-up and pull-down networks are both on during the change for a short period, the current (I_{sc}) flows directly from V_{dd} to the ground terminal and has no contribution towards charging of the output capacitance. Power consumption due to this current flow is called short-

circuit power consumption. The short-circuit power is relatively small portion compared to other power consumption [5].

In contrast to previously discussed two sources, the last source of power consumption, leakage power consumption, is due to leakage current ($I_{leakage}$) when output node is not changing. $I_{leakage}$ mainly includes subthreshold and gate leakage currents. As technology feature size shrinks, supply voltage has been scaling down to keep power consumption under control. To maintain performance, threshold voltage has been scaled accordingly. In addition to the scaling of channel lengths, to increase channel conductivity and performance, gate oxide has been scaling down as thin as possible. The reduction in threshold voltage and gate-oxide scaling increases the leakage current. In [19], the subthreshold leakage current is expressed as

$$I_{sub} = I_0 \exp\left(\frac{V_{gs} - V_t}{nV_{TH}}\right)\left(1 - \exp\left(\frac{-V_{ds}}{V_{TH}}\right)\right), \qquad (2.5)$$

where $V_{TH} = kT/q$ and $I_0 = \mu_0 C_{ox}(W_{eff}/L_{eff})nV_{TH}^2 K_{sub}$. Since V_{TH} is much smaller than V_{ds}, the term $(1-\exp(-V_{ds}/V_{TH}))$ can be neglected.

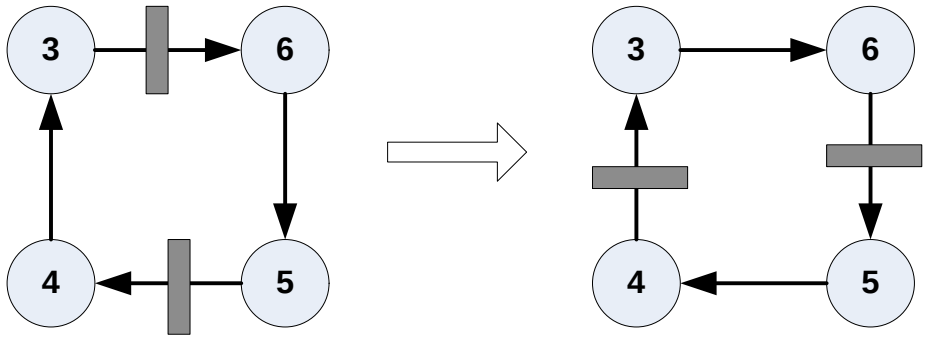
2.3 Low power/energy design

Recently, leakage power consumption has become significant with continuing technology feature size scaling. Numerous previous studies presented various device/circuit-level techniques. For example, the use of high-k material was introduced to reduce gate leakage current [20]. Various techniques such as transistor stacking, power gating, and adaptive body biasing have been presented to reduce subthreshold leakage current [21-23]. With multiple threshold voltage CMOS techniques, power gating approaches selectively power down certain blocks that are not required in the current operating mode to minimize leakage current [24-27].

For DSP applications, the reduction of switching power consumption has been the main target of low power design since switching power dominates total power consumption for computationally intensive processes. Focusing on the switching power reduction of DSP applications, numerous previous studies attempted to reduce each parameter in Equation 2.4: $\alpha_{0\rightarrow 1}$, C_L, V_{dd}, and f_{clk}. First, the choice of number representation is strongly related to switching activity ($\alpha_{0\rightarrow 1}$) of DSP. Two's complement representation is the most popular because addition/subtraction operations are easy to implement. However, when the signals do not utilize entire bit-width at most of the time (e.g., Gaussian distributed data), sign magnitude representation may result in much smaller switching activity compared to two's complement representation [6]. Hegde *et al*. introduced a MAC architecture based on sign magnitude representation for low energy consumption [11]. In addition, the switching activity can be reduced by manipulating the order of input signals. For example of shift-add multiplication, the different order of shift-add operations may result in very different switching activity [5].

In addition, strength reduction transformation in algorithmic and numerical level allows structural sharing and numerical operation sharing, which reduce capacitance (C_L) and switching activity ($\alpha_{0\rightarrow 1}$) [4]. For example, subexpression elimination (or subexpression sharing) in multiplication leads to efficient hardware implementation that consumes less power [28, 29]. Filters, DCT, and other DSP applications often contain constant multiplication. The multiplication can be divided into several additions with common subexpressions. Compared to conventional multipliers, shift-add multipliers with subexpression sharing require less computational complexity [30].

Finally, numerous previous studies attempted to reduce supply voltage (V_{dd}) based on high-level transformation techniques. Since the supply voltage level is directly related to throughput of the system, these techniques attempts to compensate the reduction in throughput while allowing supply voltage scaling. Retiming is a transformation technique that moves delay elements in a circuit to reduce critical path without affecting overall



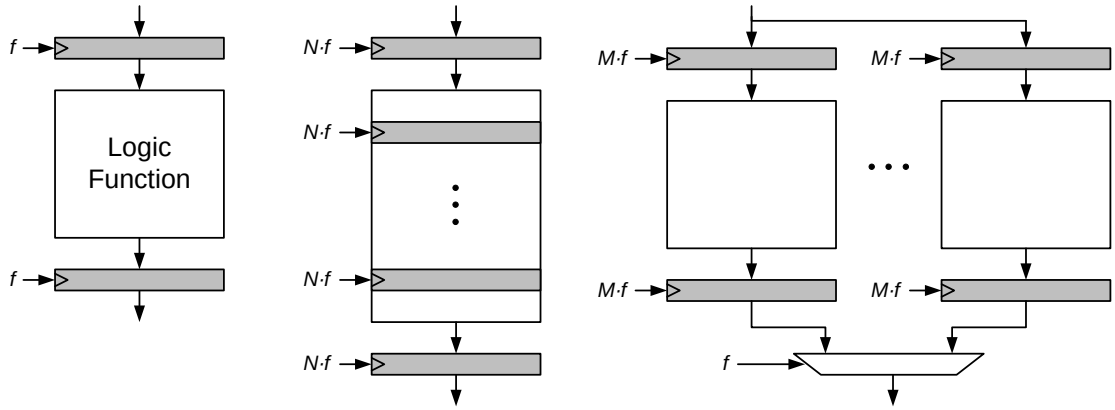
(a) (b)

Figure 2.2: Illustration of retiming with data flow graph. The number inside each circle means delay for the unit, and the gray box indicates a delay element. (a) is before retiming, and (b) is after retiming. Minimum clock period for (a) is 11, but that for (b) is 9.

computation [31-33]. Figure 2.2 illustrates a simple example of retiming. In addition, architecture driven transformation techniques such as pipelining and parallel processing relax the delay constraints on critical paths by increasing the amount of concurrency [5, 6]. Figure 2.3 illustrates pipelining and parallelization. After pipelining, the increased slack in critical paths can allow lower supply voltages at the same operating frequency, which result in low power/energy consumption. Using the parallelization approach, the duplicated units can process data simultaneously. Instead of achieving higher throughput, the parallel architecture can be operated at scaled supply voltages for low energy consumption. However, power/energy savings may be obtained through the optimal selection of the depth of pipeline and the number of parallel processing units, which are proportional to hardware complexity.

Supply voltage scaling

As discussed in Chapter 2.2, supply voltage (V_{dd}) is directly related to the all three components of power consumption. Therefore, scaling supply voltage is an efficient technique for low power/energy design since it results in quadratic reduction in dynamic power, as well as reduction in short circuit and leakage power. Under voltage scaling, the



(a) (b) (c)

Figure 2.3: Illustration of pipelining and parallelization. (a) original implementation, (b) pipelined implementation (N is number of pipelined stages), (c) parallel implementation (M is number of parallel logic function).

propagation delay increases as evident from the relationship between circuit delay (τ_d) and supply voltage given by an equation below [34].

$$\tau_d = \frac{C_L V_{dd}}{K(V_{dd} - V_t)^{\alpha}} \tag{2.6}$$

where K is a process dependent constant and α varies between 1 and 2. According to Equation 2.6, the delay increases with increasing load capacitance. The delay also increases as the difference between V_{dd} and threshold voltage (V_t) decreases. The simultaneous scaling of both V_{dd} and V_t may avert the increase in delay, but V_t scaling results in the exponential increase in static power consumption as shown in Equation 2.5. Therefore, with an optimal V_t assignment, voltage scaling is limited by the timing constraint since the increased delay may cause timing failures at critical and subcritical paths. For the non-error-tolerant environment of general purpose computing, voltage scaling often requires transformation techniques explained before or frequency scaling to prevent possible timing failures [35, 36]. Numerous previous studies presented dynamic voltage and frequency scaling techniques, which scale both voltage and frequency dependent on work load [37-40]. However, it is important to note that this approach might not be an energy efficient method for computationally intensive applications since

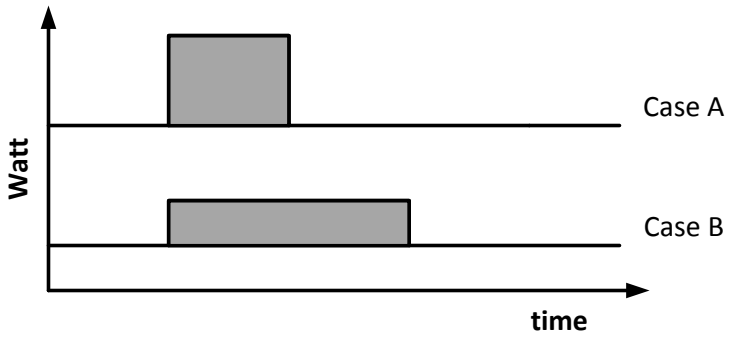


Figure 2.4: Illustration of power vs. energy. Case A is the nominal case. Case B is with voltage and frequency scaling.

energy is the integral of power consumption over a time period. As shown in the Figure 2.4, two approaches may consume the same energy though case B is considered as a low power approach. Conventionally, without performance degradation, voltage scaling has been limited to critical voltage ($V_{dd\text{-}crit}$), which ensures correct functionality [11]. Recently, aggressive voltage scaling, which reduces the supply voltage to below $V_{dd\text{-}crit}$, has been considered for the further reduction of power/energy consumption for error tolerant applications [11-14].

Relationship between voltage scalable and variation tolerant approaches

In modern digital design, timing constraints are the common factor that limits both low voltage operations and variation tolerant techniques. Process and environmental uncertainties cause timing variation and typically requires a guard band to prevent timing failures. The uncertainty in the parameters of fabricated devices from die-to-die (D2D) and within-die (WID) becomes a challenge in nanometer design [41, 42]. D2D variations affect all the devices on the same chip. WID variations may cause differences in path delay for different device on the same chip. Among the impact of process variations, channel-doping variation is the most significant one that affects threshold voltage and results in timing variation [43]. Moreover, the uncertainty in the operating environment such as temperature and noise also cannot be negligible. For example, elevated

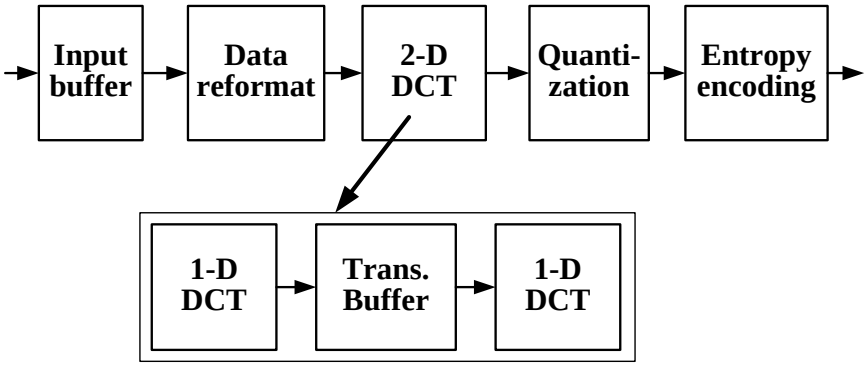


Figure 2.5: Flow graph of JPEG encoding procedure.

temperature may increase carrier mobility, so it slows down a circuit [44, 45]. For these reasons, even though this dissertation is mainly targeting low energy consumption, all discussions are very closely related to variation tolerant approaches since the source of error for the both cases is the increase in delay.

2.4 JPEG image compression

Image compression is a process that reduces the amount of data used to represent an image in order to be able to store or transmit it efficiently. One of the most popular image compression standards is JPEG [46]. The lossy compression concept of JPEG is based on the error tolerance characteristic that the human eye masks the loss in some visual information. The flow steps for the JPEG compression of grayscale images are shown in Figure 2.5. Image data are split into the contiguous sets of 8x8 blocks, and the dynamic range of pixel intensity values is converted from (0,255) to (-128,127). This step reduces the dynamic range in the discrete cosine transform (DCT) computations. The 2-D DCT converts the reformatted data to the frequency domain representation, and then quantization reduces the amount of information in high-frequency components by simply dividing each component in the frequency domain by a constant from the quantization table. The left upper corner entry of DCT coefficients is the DC coefficient, and the remaining coefficients are the AC coefficients. After DCT process, the signal power is

concentrated around the DC component. By greatly reducing the amount of high frequency information as well as reducing the overall size of the DCT coefficients, the quantization step results in a signal that is easy to compress efficiently in the encoding stage. The quantized coefficients include many zeros or small numbers, which requires fewer bits to represent with a zigzag ordering in entropy coding.

The DCT is the process of transforming a set of image samples into a set of basis sequences that are cosines. It is the most computationally expensive part of the encoding system. The 8-point 1-D DCT and the 64-point 2-D DCT are defined as follows:

$$X[u] = \frac{c[u]}{2} \sum_{i=0}^{7} x[i] \cos\left(\frac{(2i+1)u\pi}{16} \right) \tag{2.7}$$

$$X[u,v] = \frac{c[u]c[v]}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} x[i,j] \cos\left(\frac{(2i+1)u\pi}{16} \right) \cos\left(\frac{(2i+1)u\pi}{16} \right) \tag{2.8}$$

$$c[u] = \begin{cases} 1/\sqrt{2} & for \quad u = 0 \\ 1 & for \quad u > 0 \end{cases}$$

The DCT can be expressed in matrix form [47]. The 2-D DCT of an 8x8 block of image can be defined as:

$$X_{ij} = C_{ij} \cdot x_{ij} \cdot C_{ij}^{T} \tag{2.9}$$

$$C_{ij} = \frac{1}{2} \left[c[j] \cos\left(\frac{(2i+1)j\pi}{16} \right) \right] \quad i, j = 0,1,...,7 \tag{2.10}$$

The equation 2.9 can be decomposed into two separate transforms. In other words, an 8x8 2-D DCT can be implemented with two 1-D DCT's. One of them computes column-wise 1-D DCT, and the other does row-wise 1-D DCT. Figure 2.5 shows the typical VLSI implementation of 2-D DCT with a transpose buffer.

Numerous fast algorithms for DCT have been presented in the literature. In general, the efficiency of the algorithms is proportional to the number of arithmetic operations. The fast algorithms mostly attempt to reduce it using symmetry properties of the cosine basis functions. Table 2.1 shows the number of arithmetic operations for some

Table 2.1: Fast DCT algorithms

Algorithm	Multiplications	Additions
Ligtenberg[48]	208	464
Chen[49]	256	416
Chan[50]	144	464
Cho[51]	112	472
Arai[52]	80	464

popular algorithms. Arai *et al.* presented an algorithm that produces scaled results. Subsequent quantization step can absorb the scaling factor in the case of JPEG. The algorithm requires very small number of operations compared to the blind computation of equation 2.9 (1024 multiplication and 896 additions), making the algorithm one of the most efficient DCT [53].

2.5 Image quality metrics

2.5.1 Conventional metrics

Mean squared error (MSE) is the most popular assessment of signal quality and fidelity. To quantify the degree of similarity between two signals, MSE simply measures the amount that differs from reference value. For image processing, this conventional signal quality metric, MSE, is often converted into peak signal-to-noise ratio (PSNR). The calculations of the metrics are shown in the equations below.

$$MSE(I') = E[(I - I')^2] = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}(I(x,y) - I'(x,y))^2 \tag{2.11}$$

$$PSNR = 20 \times \log_{10}\left(\frac{255}{\sqrt{MSE}}\right) \tag{2.12}$$

where I and I' are M×N images, and I is a reference image and I' is a distorted image. Equation 2.12 is for the case of 8-bit images. 255 is the maximum possible pixel value. These metrics has been widely used because of its simplicity and clear physical meaning. The MSE only requires one multiplication and two additions per sample. It clearly gives information about the energy of error signal.

2.5.2 Perceptual image quality metrics

Compared to conventional quality metrics discussed above, perceptual image quality assessment is based on the characteristics of the human visual system. Several perceptual image quality metrics have been presented [54-57]. In this section, we briefly introduce one of the most popular perceptual image quality metrics, the mean Structural SIMilarity index (MSSIM), proposed in [54], which will be used extensively in this work. The MSSIM is based on the measurement of structural similarity between a distorted image and its original image as well as luminance and contrast. The underlying assumption of the structural similarity approach is that the human visual system is highly adapted to extract structural information from visual scenes. For two visually identical images, the MSSIM is 1. The quality degradation of an image results in a smaller MSSIM index. Equations shown below summarize the calculation of MSSIM index.

Mean intensity: $$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{2.13}$$

Standard deviation: $$\sigma_x = \left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_x)^2\right)^{\frac{1}{2}} \tag{2.14}$$

Correlation: $$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) \tag{2.15}$$

Contrast comparison: $$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{2.16}$$

Luminance comparison: $l(x,y)=\frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_1}$ (2.17)

Structure comparison: $s(x,y)=\frac{\sigma_{xy}+C_3}{\sigma_x\sigma_y+C_3}$ (2.18)

$$SSIM(x,y)=[l(x,y)]^{\alpha}\cdot[c(x,y)]^{\beta}\cdot[s(x,y)]^{\gamma}$$
$$=\frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)}\quad , \tag{2.19}$$

where *C3*=*C2/2* and $\alpha=\beta=\gamma=1$

$$MSSIM(I,I')=\frac{1}{M}\sum_{j=1}^{M}SSIM(x_j,y_j) \tag{2.20}$$

In Equation 2.19, the parameters are set to simplify the expression. The SSIM index is a function of two images, x and y. One of them is the reference image. The MSSIM index is simply the mean value of SSIM index.

2.6 Error tolerance of DSP applications

As a way to continue device scaling and dramatically reduce the costs of manufacturing, verification, and test, the International Technology Roadmap for Semiconductors talked about "relaxing the requirement of 100% correctness for devices and interconnects." [1] Based on the statement, the concept of error tolerant system has been adopted for numerous studies in low power design. It is apparent that the concept also can be applied for improving performance of systems, reducing manufacturing cost, building more reliable systems, and so on. For example, Breuer *et al*. introduced this concept as a new paradigm for dealing with process variations, defects, and noise [58, 59]. This dissertation is focused on efforts for reducing energy consumption based on the concept of error tolerance.

As illustrated in Figure 2.6, three dimensional design spaces (performance, power/energy, and quality) are considered for digital signal processing applications.

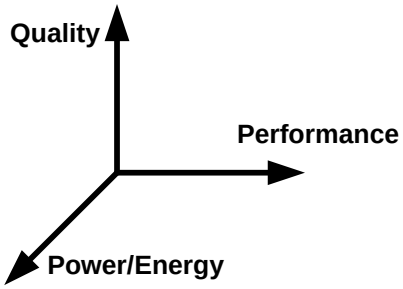


Figure 2.6: Three dimensional design spaces for DSP system design.

Performance-power tradeoffs such as dynamic voltage and frequency scaling are a popular design approach for reducing power consumption. For a high performance or low power DSP hardware design, quality-performance or quality-energy tradeoffs have been widely adapted because of the inherent error tolerance of DSP applications [11-17, 60, 61]. The output quality of DSP applications is often evaluated by the imperfect human perceptual systems, which masks small errors. For instance, the contrast sensitivity function chart in [62] shows the limit of human visual system. The contrast sensitivity drops significantly above a certain frequency. Moreover, signal visibility is strongly related to the luminance and the frequency of background signals. In other words, some error signals can be masked off when they have similar frequency, spatial location, or orientation with the background image [63]. Therefore, the error tolerance from the characteristics of the human perceptual system allows us to relax the accuracy of an algorithm and its implementation to improve performance or energy consumption. Computational accuracy can be controlled in different levels of design such as algorithm, architecture, and circuit design. For example, sampling rate, word length, or the precision of computation can be traded off for reducing hardware complexity, which is directly related to performance and energy consumption. Examples include the use of fixed point arithmetic instead of floating point arithmetic and the use of reduced bit-width for a floating point arithmetic unit [4][64]. Based on the analysis of the impact of each

computation in algorithm on output quality, some computations that are mostly redundant or result in very small effects on final output quality may be eliminated or skipped dependent on operating conditions [65, 66]. In addition, for motion estimation hardware, a bit-truncation method has been introduced to reduce the switching activity of arithmetic units [67, 68]. The truncation of several bits from the least significant bit by masking the bits to zero effectively reduces energy consumption without substantial impacts on the final picture quality. In [69], the accuracy of a filter is traded off for energy savings by adaptively altering filter order based on the signal statistics.

2.7 Prior arts in low energy techniques based on aggressive voltage scaling

2.7.1 Probabilistic computing

As opposed to traditional deterministic computing, the probabilistic computing attempts to relax rigorous accuracy constraints that are required for deterministic computing so that it can allow randomness to circuit behavior while saving energy consumption [70, 71]. As a result, various sources of randomness such as noises and parameter variation cause errors in computation. George *et al.* presented a study of the tradeoff between the correctness of arithmetic primitives in DSP applications and energy consumption [72]. In the paper, aggressive voltage scaling is applied to reduce energy consumption. Under aggressive voltage scaling, noise induces errors since voltage scaling alleviates an existing guard band for expected variations. Since the susceptibility of error depends on the voltage level, they applied biased supply voltages to reduce the impact of error on the final output quality as illustrated in Figure 2.7. Especially for arithmetic computations, higher voltages are assigned to higher order bits and lower voltages are assigned to lower order bits since errors in high-order bits cause a large magnitude of error in the computation and result in significant impact on final output quality. The major issues of this technique are how many different voltages can be allowed in a fairly

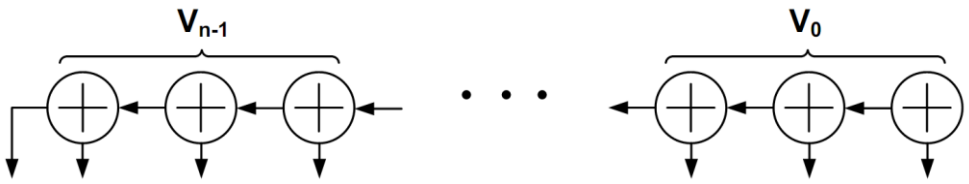


Figure 2.7: Illustration of biased voltage scaling for a ripple carry adder. In this case, the number of different voltages is *n* and $V_i > V_{i-1}$, where $1 < i < n-1$.

small region. In addition, multiple voltage design requires independent power supply and grid structure and additional level converters inside the arithmetic units, which increase area and power overhead.

Based on the similar background of error tolerance and multiple voltage design, Chakrapani *et al.* presented a mathematical model for accuracy-energy tradeoffs [73]. The main difference of this analysis from previous one is the assumption that the source of errors is increased propagation delay instead of noise under aggressive voltage scaling. The analysis and experiments of the two previous studies are based on the delay estimation using the carry propagation behavior of arithmetic units.

2.7.2 Error compensation based on inherent error tolerance

Based on the inherent error tolerance of DSP applications, numerous techniques attempt to permit erroneous operations under aggressive voltage scaling and then compensate the errors using additional error compensation logic. Hegde *et al.* proposed a design methodology, algorithmic noise-tolerance, to reduce the degradation in output quality using separate error control logic as shown in Figure 2.8 [11][12]. Errors are detected by comparing the output of main processing block ($y_a[n]$) under scaled supply voltage to that of estimator block ($y_p[n]$). In order to compensate the delay error under aggressive voltage scaling, the output from the estimator substitutes for the erroneous output when the difference between the outputs from the two blocks exceeds a predefined threshold. The estimator can be a reduced precision replica of the main block or a predictor based on the past output of the main block [14][15]. The estimator does not

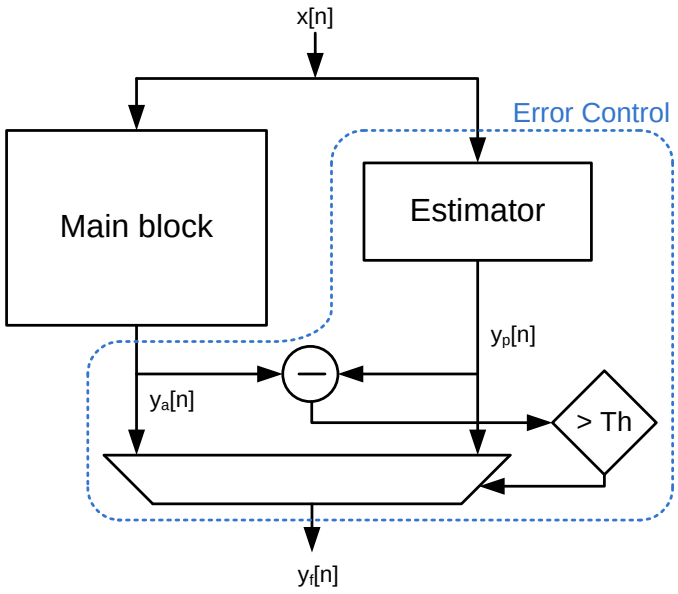


Figure 2.8: Algorithmic noise tolerance scheme.

provide 100% accurate results. However, somewhat inaccurate operations may be acceptable for many DSP applications in terms of inherent error tolerance of the applications. Based on the algorithmic noise-tolerance scheme, several approaches have been presented for various DSP applications [11-17]. The main issue of this approach is overhead from the additional hardware requirement for the error compensation steps. The redundant unit needs to be relatively smaller than original unit to minimize the possible energy consumption penalty.

Kurdahi *et al.* presented a study of memory errors under aggressive voltage scaling for DSP/communication applications and proposed an error compensation technique based on a complex iterative error concealment algorithm [74]. The error compensation algorithm implemented in the decoder detects and compensates the erroneous output caused by voltage scaled memory in the encoder. This is based on the assumption that an image is encoded in mobile devices and decoded in devices that does not need low-energy consumption.

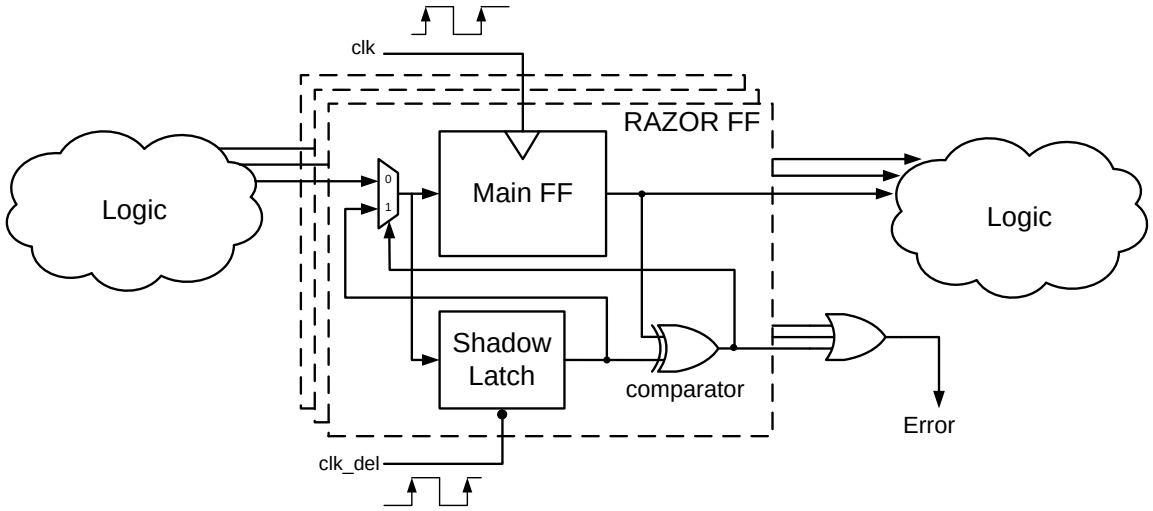


Figure 2.9: RAZOR flip-flop architecture.

2.7.3 In-situ timing speculation and error control

Conventional dynamic voltage scaling methods modulate the supply voltage dependent on performance demand. In contrast, the RAZOR approach presented in [18] provides a feedback based voltage control scheme based on error rate. The approach includes in-situ timing error detection using a shadow latch, which captures a signal at the negative edge while a normal flip-flop does at the positive edge. Figure 2.9 shows the RAZOR flip-flop architecture. In the presence of a timing failure due to increased delay under aggressive voltage scaling, the shadow latch still can capture correct value. Therefore, by comparing the captured signal at an original flip-flop to that at the shadow latch, timing failures can be detected easily. Whenever an error is detected, correct computation is achieved using replay mechanisms in pipelined micro-architectures. This approach is based on the assumption that the error occurrence is not very frequent and highly data dependent since it is directly related to performance degradation. By monitoring error rate, the supply voltage is controlled dynamically to achieve optimal energy savings with minimum performance penalty. Since this error control technique is focused on pipelined micro-architectures and provides error-free results, this technique is

suitable for general purpose computing architectures. Recently, RAZOR II in [75] reduces the complexity and size of the RAZOR flip-flop, which is related to energy overhead.

Fuketa *et al*. presented a method to predict timing failures using additional flip-flops called Canary flip-flops [76]. The Canary flip-flop, which includes a delay buffer, causes timing errors when the timing margin becomes smaller than a predefined value. A warning signal for timing failures is generated by comparing the value from the Canary flip-flop to that from a regular flip-flop. This timing error prediction approach does not require error correction or compensation. Instead, additional speed control circuit controls system performance based on the generation of the warning signal.

2.7.4 Critical path modulation and clock stretching

Under voltage scaling or process variation, short paths may not experience timing errors while long paths such as critical paths violate timing constraints. The CRISTA approach presented in [77] attempts to isolate the critical paths from non-critical paths with a circuit partitioning method using hierarchical Shannon expansion and gate sizing. When the isolated paths are exited, extra clock cycle is allowed to prevent delay errors as illustrated in Figure 2.10. This adaptive clock stretching requires additional latency prediction block to determine if the current computation requires clock stretching or not [77, 78]. The latency predictor block generates an enable signal that control the clock signal for registers before the next set of inputs arrives. Since the two cycle operation causes performance degradation likewise the RAZOR approach explained above, this method needs to ensure that the excitation of the isolated paths is not very frequent after the path modulation.

Banerjee *et al.* introduced a design methodology that adaptively selects the critical path dependent on the voltage level [79]. Based on the algorithmic transformation that distinguishes significant computations from non-significant computations, critical path

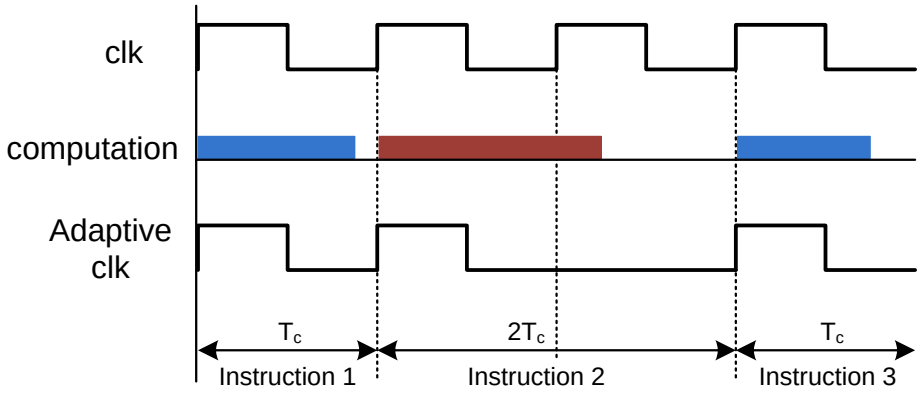


Figure 2.10: Timing diagram of adaptive clock stretching.

simply altered using multiplexors to ensure that output does not include erroneous results under the prespecified voltage level. By selecting shorter path under scaled voltage, timing failures are precluded with the cost of quality degradation due to imprecise computations.

2.7.5 Dynamic timing control based on time borrowing

Compared to the hard boundary of flip-flops, latches have transparency for a half clock cycle. Since flip-flops captures signals only at clock edge, data should be arrived before the clock edge. In contrast, for latches, data can be arrived anytime within the transparent window. This property can be useful when a system experiences variations in data arrival. However, because of large transparent window, latch based design may have serious hold time violation issues, so it is not mostly adapted by standard EDA tools. Several prior studies presented flip-flop designs that intentionally generate transparent window for variation tolerant design. Joshi *et al.* discussed the flip-flop design that allows the time borrowing window, which is called soft-edge flip-flop (SEFF) and introduced an algorithm to assign SEFFs for maximum delay improvement with minimum power overhead [80]. Chae *et al.* used SEFFs to allow low voltage operations [81]. When time borrowing is occurred under a scaled voltage, slack for next stage operation is reduced by the borrowed time. In order to pay the borrowed time back, clock

shifter stretches the clock period for the next clock cycle. This approach allows voltage scaling within the range of time borrowing while trading off the system throughput.

CHAPTER 3

ERROR ANALYSIS UNDER SCALED VOLTAGES

3.1 Introduction

Traditionally, the critical voltage ($V_{dd\text{-}crit}$) has been determined by estimating the critical path delay based on the static timing analysis. $V_{dd\text{-}crit}$ is simply the minimum voltage that satisfies a desired timing constraint. Without any approaches that reduce the critical path delay, erroneous operations due to timing failures are expected under the voltage level lower than $V_{dd\text{-}crit}$. However, this corner based approach may be overly pessimistic, for the worst-case corner rarely occurs. Several previous work indicates that the critical path is excited only with certain inputs, thus the circuit may often operate correctly even under $V_{dd\text{-}crit}$ [11, 14, 18]. For any possible approach based on this property, accurate error analysis is the fundamental step. Since the error arises from timing failures due to increased delay under aggressive voltage scaling, the error analysis requires accurate delay estimation.

For the error analysis for additions, which are the most frequent operation in DSP application, the analysis of carry propagation in addition has been used to model path delay in numerous prior work [11, 72, 73, 77]. However, carry propagation does not provide a true value of delay error for continued additions because it does not consider actual delay value based on transitions at nodes. Hence, in this chapter, transition based delay estimation is introduced to obtain accurate delay value at each output bit.

This chapter first discusses the circuit behavior under aggressive voltage scaling and then explains the difference between input dependent delay estimation methods. With detailed comparison between the two different delay estimation schemes and the comparison results of error rates under aggressive voltage scaling, it describes the

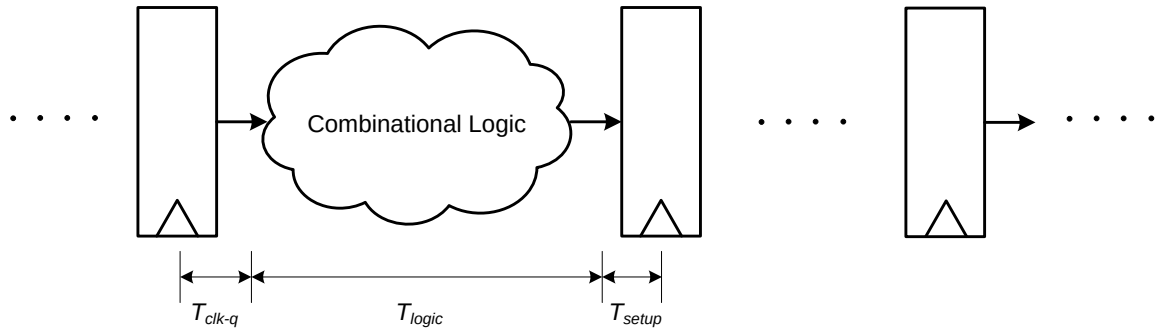


Figure 3.1: Illustration of operating delay calculation in pipelined architecture

importance of sequence dependent delay estimation. Then, we explain the simulation framework for the error analysis widely used in this work. This chapter also introduces an event based error model using a finite sate machine and explains how it can be expended to various arithmetic units.

3.2 Impact of voltage scaling on propagation delay

In general, each pipeline stage has logic components bounded by timing elements such as flip-flops as shown in Figure 3.1. To set an operating period (T_{oper}=1/f_{oper}), the maximum logic delay (T_{logic}) of logic component at the nominal voltage is measured. Then, the operating period is calculated as shown in below.

$$T_{oper} = T_{logic} + T_{clk\text{-}q} + T_{setup} \tag{3.1}$$

where $T_{clk\text{-}q}$ is the clock to Q delay of a flip-flop and T_{setup} is the setup time. If the delays of logic and timing elements under various conditions are increased and the sum of the delays is greater than T_{oper}, correct values may not be captured at the timing elements.

When the timing failure is considered as the main source of erroneous output, the variation in delays for both logic and timing elements is the most significant factor. Figure 3.2 shows the average delay results for a 1-bit fulladder (FA), a latch, and a flip-flop with respect to different supply voltage levels. This simulation considered two different technology nodes, which are 180nm and 45nm predictive technology models (PTM**)** [82] with voltage scaling factor in the range of 0.5 ~ 1 (0.9V ~ 1.8V for the

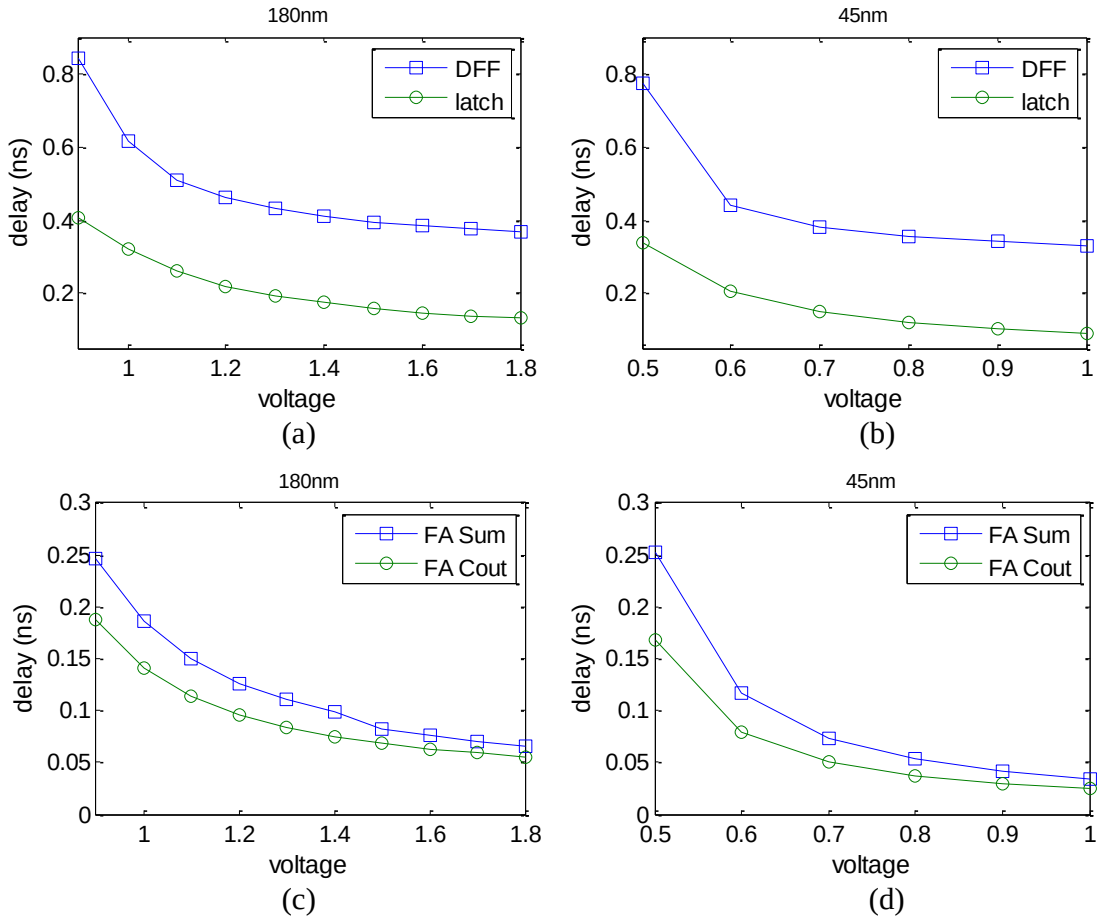


Figure 3.2: Average delay results for basic components: (a) a latch and a flip-flop with 180nm technology, (b) a latch and a flip-flop with 45nm technology, (c) a 1-bit fulladder with 180nm technology, (d) a 1-bit fulladder with 45nm technology

180nm technology and 0.5V ~ 1V for the 45nm technology). The FA simulation is based on 28 transistor design [8]. The flip-flop and latch designs are obtained from the standard cell library of each technology. The detailed spice simulation setup is explained in Chapter 3.4. For all three components, the experimental results show that the propagation delay is exponentially increased as the supply voltage is scaled. We can intuitively figure out that the delay increase in timing elements causes relatively very small impact on the extent of total delay increase since they occupy small portion compared to entire logic in a pipeline stage; the delay increase for only one bit timing element is added to final delay. Therefore, remaining sections focus on delay increase in logic, especially arithmetic units which are the main components in DSP systems.

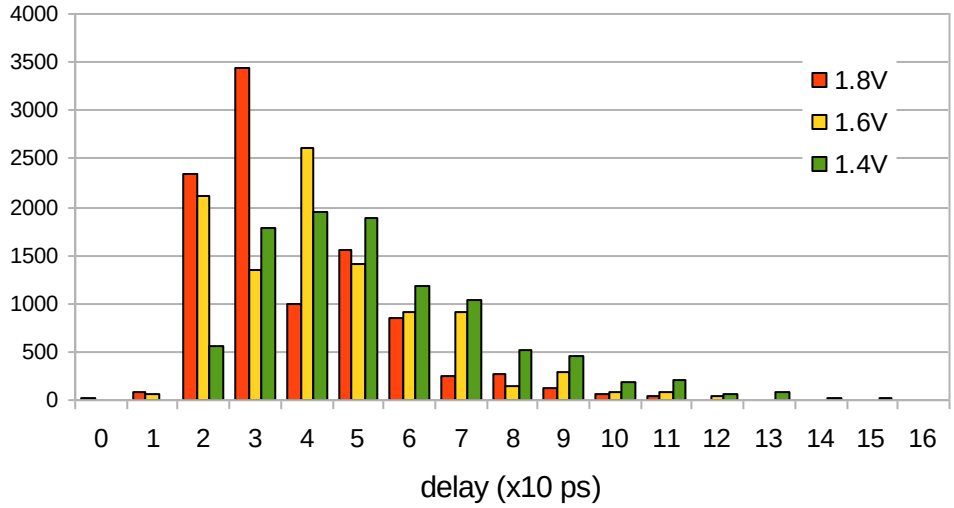


Figure 3.3: Example delay distribution of an 8bit adder.

Figure 3.3 shows the delay distributions of an 8 bit ripple carry adder with respect to three different supply voltage levels. This experiment uses 10000 normal distributed input vectors with a zero mean and the standard deviation of 50. The maximum delay for each input vector is measured using NanoSim [83]. As illustrated in the figure, delay widely varies dependent on input vectors. In addition, voltage scaling increases the variance of the asymmetric distribution. These properties show that the occurrence of a delay error is highly input dependent. Therefore, the input dependent delay estimation is important for accurate error analysis, especially for arithmetic units. Next section will discuss different delay estimation methods and compare them.

3.3 Static path delay-based estimation vs. transition delay-based estimation

3.3.1 Static path delay-based estimation

Using static path delay-based scheme, the delay estimation of a ripple carry adder (see Figure 3.4) is based on carry generation and propagation. As shown in the figure, a FA includes three input bits: *A*, *B*, and *CarryIn*, and two output bits: *Sum* and *CarryOut*. A bit position generates a carry if $A_i=B_i=1$ and propagates a carry if either A_i or B_i equals

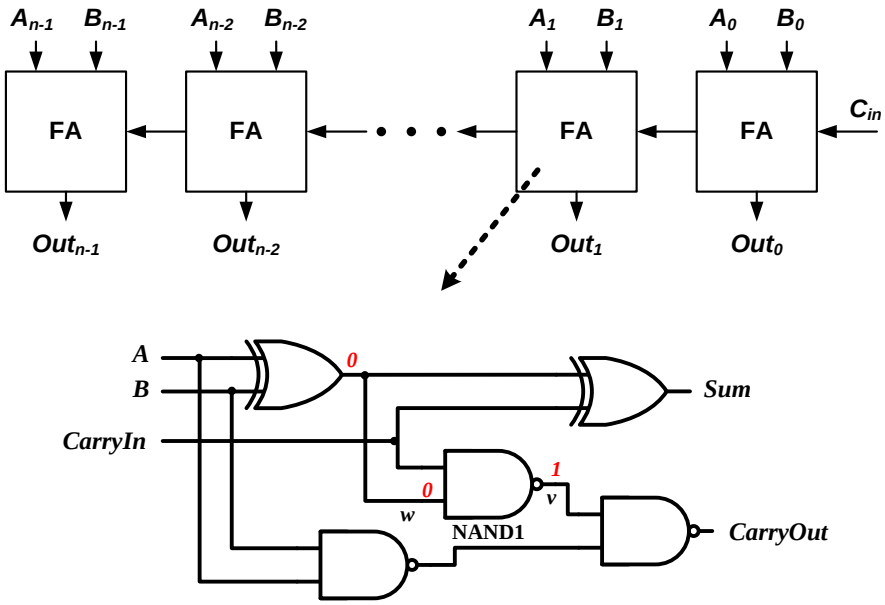


Figure 3.4: Ripple carry adder structure and full adder gate level implementation

1. The path delay is based on the length of the longest carry chain that transfers the carries of *1* successively from low-order bits (LOBs) to high-order bits (HOBs). For example, the addition of two numbers *x* and *y*, *x = 11111111* and *y = 00000001*, results in a carry generation at the least significant bit (LSB) and propagates the carry to the most significant bit (MSB). This case causes the longest carry propagation path and excites the critical path. If *x = 11101111* and *y = 00000001*, the carry propagation stops at the fifth bit position and no carry propagates to a higher order bit position. The delay for the inputs includes only five FAs from LSB. This scheme often ignores the delay variation of each FA due to different input conditions and uses fixed delay value for that. More importantly, delay estimation is dependent on only current input combinations using this scheme.

3.3.2 Transition delay-based estimation

Compared to the static path delay-based estimation, transition delay-based estimation focuses on the switching activity of output bits. For this scheme, delay is

generated whenever the transitions of inputs cause transitions at CarryOut and Sum. For an extreme case, if the previous input vector and current input vector are same, output will not be changed. In this case, the delay is zero. Dependent on previous and current input vectors, some part of input/output may not have transition activities likewise the extreme case, and this affects the propagation delay. Therefore, for transition delay-based estimation, delay is based on the relationship between previous input vector and current input vector. Compared to delay generation, delay propagation is determined by only current input vector. A FA only transfer the generated delay to the adjacent higher-order full adder if the two current inputs to a full adder, A_i and B_i, are not equal. As shown in the gate-level diagram of the FA (see Figure 3.4), if $A_i=B_i$, w is always 0 and v is always *1*. In this case, *CarryOut* is determined by only A_i and B_i without respect to *CarryIn*. Consequently, the n-bit ripple carry adder can be divided into independent subsequent parts that can operate in parallel. For example, if input to the adder is $x = 11110111$ and $y = 00000000$, 4^{th} input bits are the same. Thus, the maximum delay can be up to the value for four FAs since the upper four FAs does not need *CarryIn* value from lower four FAs. That is, the number of consecutive $A_iB_i = 10$ or 01 inputs determines the length of delay propagation. By considering the both delay geneation and propagation, the transition delay-based method may result in accurate delay estimations.

The example shown in Figure 3.5 illustrates the consecutive additions to clearly show how the transition delay-based estimation scheme differs from the static path delay-based estimation schme and how the differences lead to very different delay results. When the second input combination is the current operation, the static path delay scheme considers it a maximum delay case as explained above. For the transition delay-based scheme, the second input combination also results in a long delay propagation path (long consecutive *10* inputs), but the actual delay will be very small because the *CarryOut* bits of each full adder do not have any transition activities (from 1^{st} to 2^{nd}). Therefore, the case will have no error under aggressive voltage scaling. When the third combination is

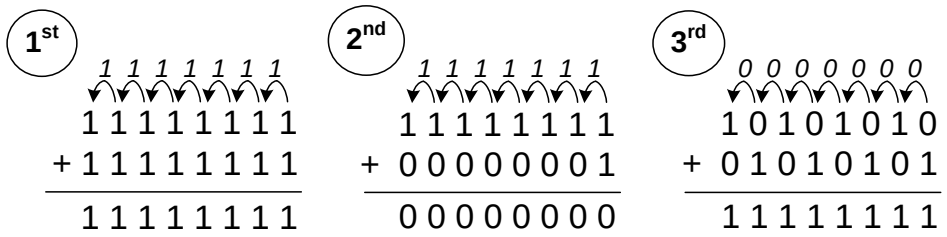


Figure 3.5: Example of a sequence of 8-bit addition.

the current operation, the carry propagation scheme considers it a minimum delay case since there is no carry propagation. However, it is considered a long delay propagation case for the transition delay-based scheme, and the actual delay may be large enough to cause delay faults since *CarryOut* for all FAs have falling transitions (from 2nd to 3rd).

3.4 Simulation framework and comparison results

3.4.1 Background

The error analysis is based on a simple structure in which an input vector from input registers are inserted to an arithmetic unit and the calculated value is captured at output registers. It is defined that an output bit has a delay fault if the delay at the output bit is greater than the operating delay (T_{oper}). The analysis assumes that the output register holds the previous value when the output bit has a delay fault. This assumption is valid when violating a setup time does not put the latch in a metastable condition and the register is not reset between successive additions. Note that the previous output bit value can be the same as the correct output bit value. In this case, the output is correct even though the input combination causes a delay fault under voltage scaling. Therefore, an output bit is erroneous only if the captured value is different from the correct value.

3.4.2 Transition delay-based estimation framework

A circuit delay calculation based on data-dependent gate delays proposed in [84, 85] requires gate-level delay modeling. However, if the size of a circuit is large, the gate-

level approach consumes tremendous simulation time. Moreover, it is not efficient to examine timing failures since the delay at output is the only concern. Thus, a FA is the smallest component in the simulation model. First, the delay values of the basic component, a 1-bit FA, are obtained for all possible input sequences using HSPICE simulations [83]. Then, using the data from the simulations, behavioral C simulation calculates the delay values for each output bit of the *n*-bit adder based on a current input sequence. The 1-bit FA simulation used the 28-T full adder design with a load of three minimum size inverters and PTM 70nm technology. The simulation includes estimations of delay and energy consumption of the full adder for all possible input transitions and different voltage levels. The voltage scaling factor is set in the range of 0.67 ~ 1 (i.e. 0.8V~ 1.2V). Three inputs of a FA generate 64 possible input transitions and corresponding delay values to each output.

As illustrated in Figure 3.6, the behavioral C simulations examine the presence of delay faults and errors for each output bit. The final delay for each output bit is calculated based on the transition delay-based estimation method explained in Chapter 3.3.2. If a calculated final delay value is greater than a predefined operating delay, it is counted as a delay fault at the output bit. When a delay fault occurs, the value of the output bit is compared with the correct value to determine if the bit position is erroneous or not. This process is repeated for all *n* FAs.

Likewise the delay estimation, the energy consumption of the adder for each input sequence is obtained based on the HSPICE energy estimation results of a full adder for all 64 input transitions. For each input sequence, current drawn from power supply is measured and used for calculating energy consumption. The total energy consumption is simply the sum of all energy consumption of each component dependent on input vector.

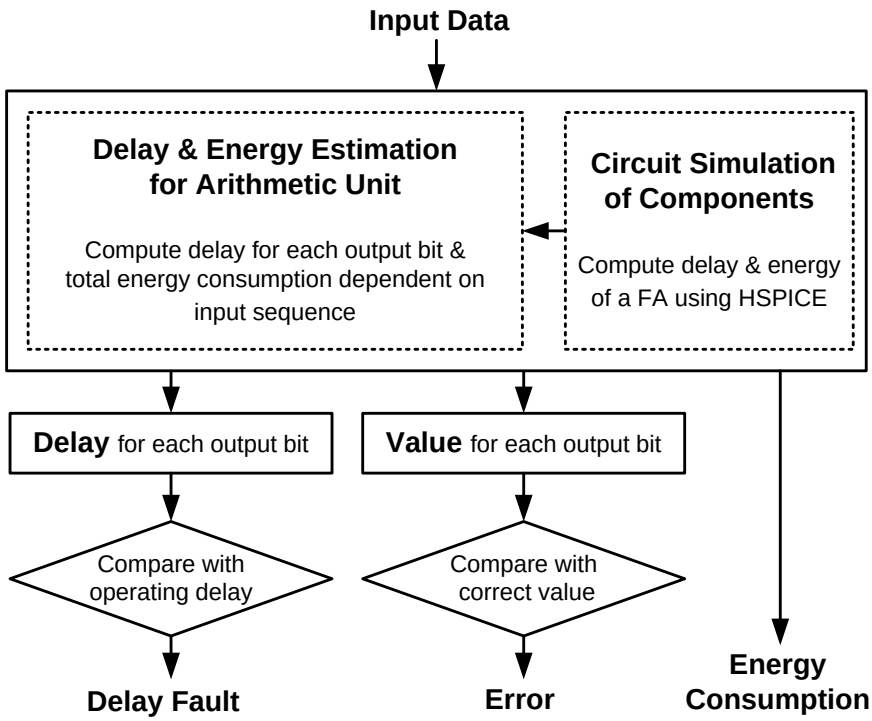


Figure 3.6: Flow chart of behavioral C simulation for error and energy analysis

3.4.3 Comparison results of delay estimation methods

With experimental results, this subsection compares three delay estimation schemes: worst-case delay estimation, static path delay-based estimation, and transition delay-based estimation. For the worst-case delay estimation, the delay of a path is the sum of the worst-case delay of each gate on the path without respect to the inputs to the circuit. We can intuitively figure out that there is always a timing failure at the MSB whenever the voltage results in a delay greater than a given operating delay. The voltage level determines the number of faulty bits from the MSB toward the LSB. Thus, the error rate is only dependent on the probability that a current erroneous bit is not equal to the correct value. Figure 3.7 and 3.8 show the comparison results of the timing failure rates for three arithmetic units: a ripple carry adder, a Kogge-Stone adder, and an array multiplier. For the experiments, we used 10,000 normal random input combinations with a zero mean and a standard deviation of 10. Compared to the ripple carry adder, the

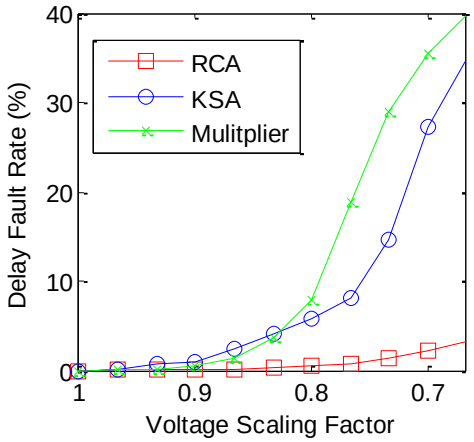


Figure 3.7: Delay fault rate of MSB for different arithmetic units.

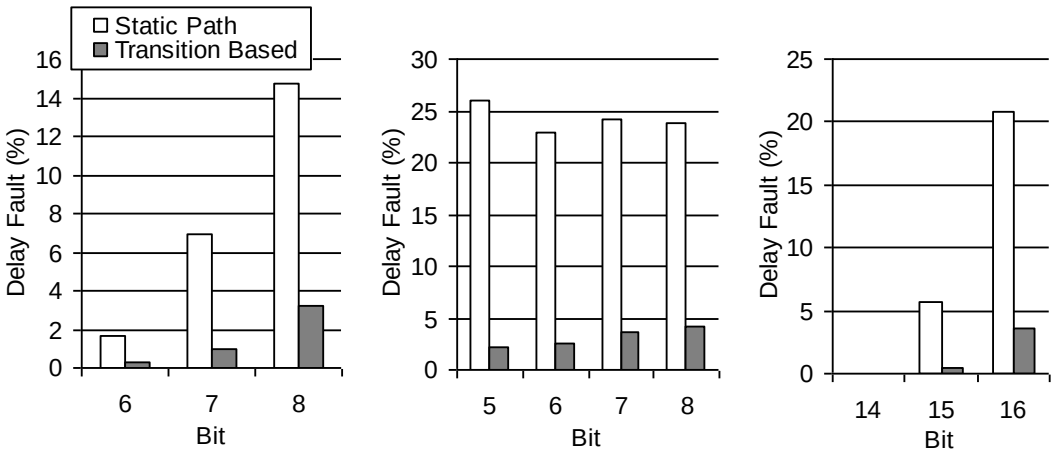


Figure 3.8: Delay fault rate comparisons for different arithmetic units: (a) a 9-bit ripple carry adder, (b) a 9-bit Kogge-Stone adder, and (c) an 8x8 array multiplier (no errors at low-order bits).

Kogge-Stone adder and the array multiplier were very sensitive to voltage scaling and resulted in a high delay fault rate with a small decrease in voltage because their parallel architectures include many subcritical paths and reduce the timing slack between the critical and subcritical paths. For the comparisons of the delay fault rate among the different delay estimation schemes, the voltage scaling factor for the ripple carry adder was 0.67, and that for the Kogge-Stone adder and the array multiplier was 0.83. Figure 3.8 does not include the results for the worst-case delay estimation scheme since the

delay error rates were always 100%. Based on the results of the comparison, the transition delay-based estimation results in smaller delay fault rates than the other schemes. The huge differences among the three schemes show that accurate delay estimation is very significant for an analysis of correct circuit behavior under aggressive voltage scaling.

3.5 Error models for arithmetic units

This section presents a general delay estimation model based on transition event, using a finite state machine (FSM). Since delay generation depends on the transition event from a previous to a current input, the modeling of delay can be expressed using the transition of states. For a ripple carry adder, the nodes that need to be monitored are the *CarryOut* bits of each FA since they are on the critical path. Each value of a *CarryOut* bit becomes a state of the FSM, and inputs to the FA are inputs to the FSM. Figure 3.9 illustrates the case in which the value of *CarryIn* is *0*. If *CarryIn* is *1*, a completely opposite FSM is required. The output of the FSM is a transition event on *CarryOut* (whether the *CarryOut* changes from a previous to a current operation). The figure also includes the equations that describe the functionality of a FA. Note that propagation (P) and the value of *CarryOut* are determined by combinational logic, but the generation of delay at *CarryOut* is sequential logic. Let us explain the FSM shown in the figure using examples. If *CarryOut* for the previous operation ($(t\text{-}1)^{th}$) is in state *1* and inputs *A* and *B* to a full adder are (*0,1*), then *CarryOut* for current operation (t^{th}) transits to state *0*. Thus, a falling transition (T) on *CarryOut* generates delay. Based on the equation for *P*, the input combination propagates the generated delay from a previous stage (FA_{i-1}) to a current stage (FA_i). Therefore, the total delay to *CarryOut* is the sum of the generated delay and the delay from the adjacent full adder (FA_{i-1}). If *CarryOut* at $(t\text{-}1)^{th}$ operation is in state *0* and inputs *A* and *B* to a full adder are (*0,1*), *CarryOut* at the t^{th}

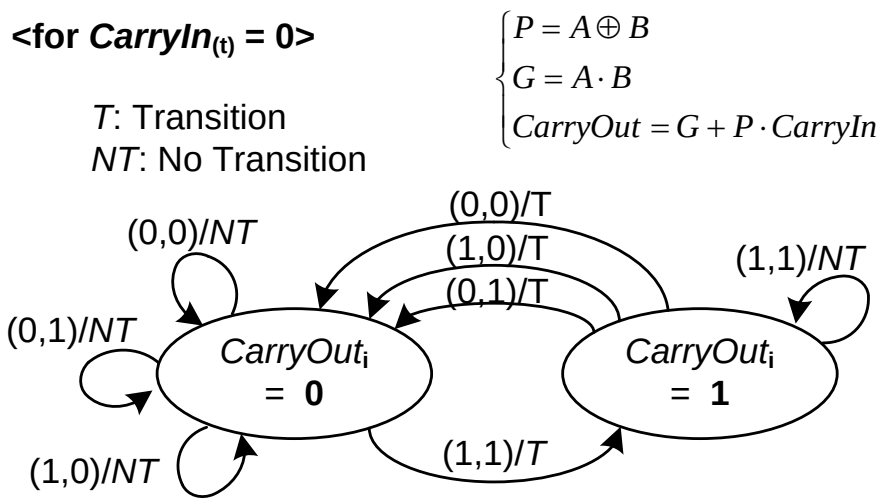


Figure 3.9: Event based finite state machine.

operation remains in state *0*. This case has no transition (NT), so propagation is not a concern.

Using this model, the presence of delay faults at each output bit can be estimated by comparing the estimated delay value to the T_{oper}. The delay at each output bit is the sum of unit delay on the path that only includes consecutive transition activities, which is proportional to the probability of error. This is summarized in an equation shown in below.

$$P_n \propto D_n = \sum_{j=k}^{n} D_{unit\_j} \tag{3.2}$$

where P_n = the probability of error at n^{th} bit, k = the starting bit position of consecutive transition activity up to n^{th} bit, and $D_{unit\_j} = j^{th}$ unit delay based on input sequence.

This model can be extended to various other adder and multiplier architectures. Carry skip adder has almost the same architecture with ripple carry adder, but it includes carry skip components as shown in Figure 3.10. For example of a carry skip adder with a fixed block width, each block follows the same model discussed above. If current inputs do not result in the carry skip condition (i.e., $P_1 \cdot P_2 \cdot P_3 \cdot P_4 \neq 1$), delay propagation through

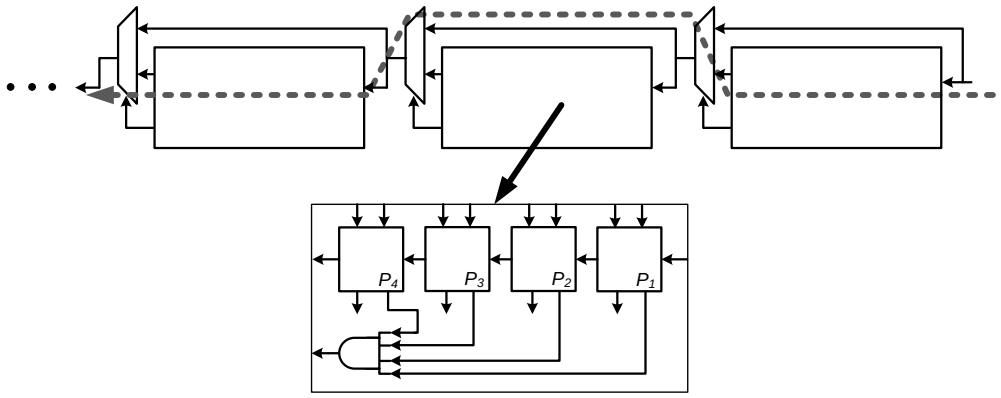


Figure 3.10: Carry skip adder structure. The dashed line indicates the critical path when the second block is skipped because $P_1 \cdot P_2 \cdot P_3 \cdot P_4 = 1$.

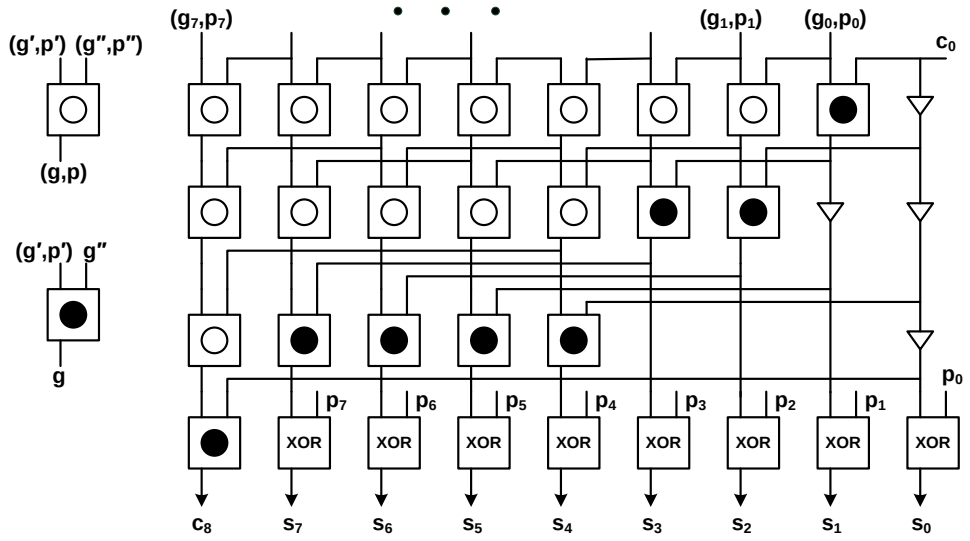


Figure 3.11: Kogge-Stone adder structure [86].

the blocks is the same with that for a ripple carry adder. However, when $P_1 \cdot P_2 \cdot P_3 \cdot P_4 = 1$, the initial delay for the i^{th} block is the sum of delay values of $(i\text{-}1)^{th}$ block and carry skip logic. For parallel prefix adders such as Kogge-Stone adder and Brent-Kung adder, the basic unit can be the carry operator, ¢ (i.e., $(g, p) = (g',p') ¢ (g'',p'')$ means $g = g''+g'p''$, $p = p'p''$, *where g=generate, p=propagate*) [86]. Figure 3.11 illustrates Kogge-Stone adder structure. The transition activity of each input (g',p',g'',p'') and output signal to the carry operator (g and p) are monitored to estimate delay faults likewise full adder model discussed above. The prefix adders include more than one critical path, and the length of

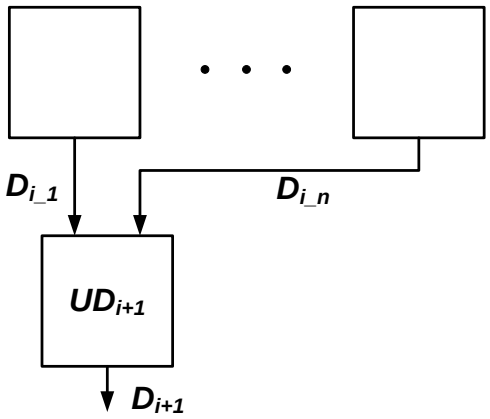


Figure 3.12: Illustration of delay propagation. UD_{i+1} is the delay for the unit, and D_{i_1}, …, D_{i_n} are the delays from the i^{th} units.

subcritical paths is not very different from that of the critical paths. In this case, classifying all possible paths and examining delay propagation through the paths may not be very efficient way to estimate delay faults on output bits. In fact, to verify the presence of delay fault, it is not necessary to identify critical path for each output bit. In addition, for accurate delay estimation, the possible paths for each output bit include almost all components in the architecture. Therefore, to reduce simulation complexity, the delay propagation for each component can be modeled using the iterative equation below.

$$D_{i+1} = Max(D_{i\_1},...,D_{i\_n}) + UD_{i+1} \tag{3.3}$$

As illustrated in the Figure 3.12, final delay (D_{i+1}) of $(i+1)^{th}$ unit is the sum of the maximum input delay and the unit delay based on input sequence. This iterative formula can be applied for the delay estimation of other parallel adders and multipliers such as array and tree multipliers. It is important to note that each unit delay for each output bit is based on the transition activity dependent on input sequence to the unit.

3.6 Summary

This chapter discusses the error analysis under a scaled voltage. After explaining the circuit behavior with voltage scaling, it demonstrates the importance of the accurate

delay estimation, which depends on not only combinational inputs but also the previous state of logic. This chapter provides the detailed comparisons among different delay estimation schemes and points out why the transition delay-based estimation should be adapted for accurate error analysis. In addition, a sequential model using finite state machines is presented for accurate error estimation.

CHAPTER 4

PERCEPTION-BASED ERROR TOLERANCE AND ENERGY SAVINGS

4.1 Introduction

Previous approaches based on aggressive voltage scaling explained in Chapter 2.4 have mostly ignored the interaction between the input signal/image and the voltage scalability of hardware. This chapter discusses the dependence of voltage scalability on input image characteristics and its implication on energy savings in error tolerant image processing. The output image quality of image processing is dependent on input image under aggressive voltage scaling due to two reasons. i) There exists natural disparity in error tolerance among images in terms of perceptual image quality assessment. ii) The error rate for logic under aggressive voltage scaling varies with input image types.

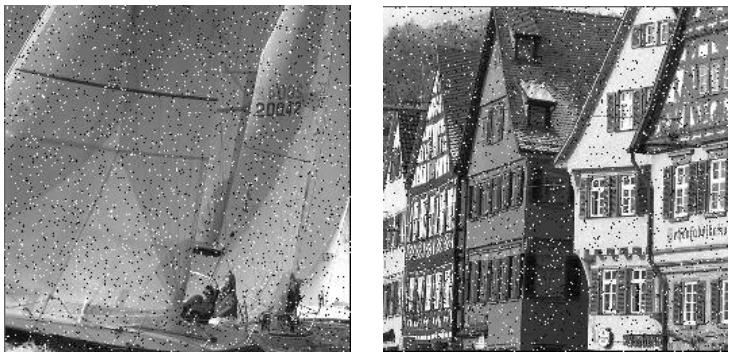
In the first, visual information is not perceived equally due to the complicated non-linear characteristics of the human visual system [63]. Some information is more important than other information in terms of quality assessment. Therefore, the quality degradation of some images is evaluated as smaller than that of other images even though the same error signal is added to the images. In the second, the error rate due to timing failures varies with different input images since the characteristics of input images excite or diminish conditions causing a timing failure. In most error tolerant DSP systems, arithmetic units (i.e., adders and multipliers) are major components that determine the system performance and its energy consumption. As discussed in the previous chapter, under aggressive voltage scaling, the excitation of long delay path in an arithmetic unit causes erroneous outputs. The error rate is directly related to the frequency of the excitation, which is dependent on input sequence.

This chapter first explains the natural disparity in error tolerance due to the characteristics of the human visual system. Then, we discuss that different input images result in very different input sequences to arithmetic computations, which vary error rate under aggressive voltage scaling. As the voltage level is determined by the output quality, we discuss the difference in energy consumption among images if input image types are examined during voltage scaling. In addition, the experimental analysis includes a validation of the results under process variation and different technology node and image size. The analysis and experiments in this chapter establishes a fundamental understanding of the relationship between inputs to error tolerant systems and energy savings.

4.2 Natural disparity in error tolerance

Since image quality is ultimately evaluated by the human visual system, subjective quality assessment is the most reliable method. However, because the subjective method is too slow for real-world applications, various objective methods have been used to access image quality. This section explains the fundamental problem of conventional image quality metrics and the difference between conventional and perceptual image quality metrics. Then, it discusses the existence of disparity in error tolerance among different images based on perceptual image quality assessment.

Conventional image quality metrics such as the mean squared error and the peak signal noise ratio consider only the strength of an error signal as discussed in Chapter 2. The metrics have been widely used because of their clear physical meaning and simplicity. However, conventional metrics poorly correlate with perceived image quality since the visibility of errors is dependent on both the strength of an error signal and the characteristic of the human visual system. Within the limit of grayscale image processing, three aspects of the human visual system—contrast masking, texture masking, and frequency masking—help to explain the concept of perceptual image quality assessment.



(a) (b)

Figure 4.1: Example comparison of two images with same salt & pepper error.

If the visibility of errors is considered the effect of an error signal on the background image, the extent of contrast, texture, and frequency of the background image changes the discrimination threshold [63]. For example, with applying the same error signal to two different images, the human visual system generally detects less error in the images with high contrast, strong texture, and numerous high frequency components. In contrast, the error signal is more visible in the images with low contrast, less texture, and fewer high frequency components since the characteristics of the images increase the discrimination threshold.

The experimental analysis use one of the most popular perceptual image quality metrics, the Mean Structural SIMilarity index discussed in Chapter 2. To show the difference between conventional and perceptual image quality assessment, the same randomly distributed salt and pepper error to two different images is applied (see Figure 4.1). The PSNR results for two images are 21.5dB, but MSSIM results differ by about 48% (0.5443 for (a) and 0.3833 for (b)). Image (b) has more strength in all the aspects—contrast, texture, and frequency—than image (a). With the same error signal, more structural information is also preserved in image (b) than image (a).

Therefore, based on the characteristics of images, the perceptual image quality evaluated by the human visual system may be very different with the same strength of an error signal due to the correlation between an error signal and an image. In other words,

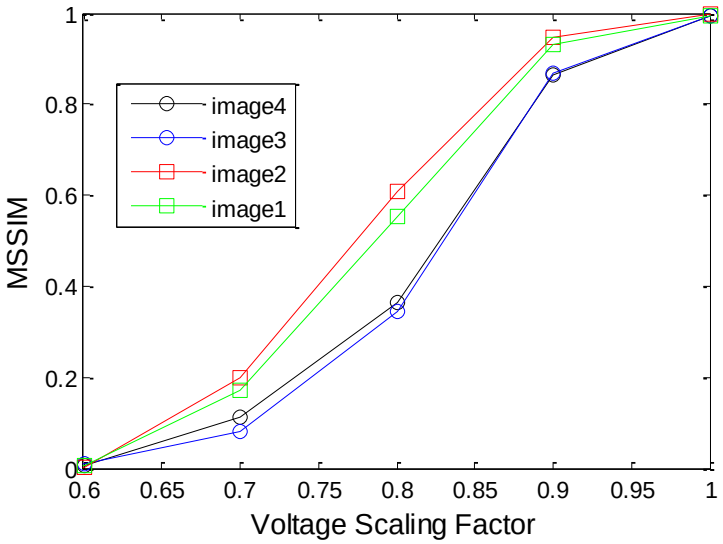


Figure 4.2: Comparison of image quality after memory error.

images with higher contrast, texture, frequency, and structural information naturally tolerate more error signals in images. One of the main concerns of the natural disparity in error tolerance among images is how accurately to evaluate image quality. Since this research does not focus on studying or developing methods for image quality assessment, the experimental analysis relies on the accuracy of the MSSIM index.

Any part of an image processing system that allows erroneous outcome may utilize the natural disparity in error tolerance to optimize the energy consumption or reduce the quality degradation. Figure 4.2 shows the example comparison of output quality for different images when aggressive voltage scaling is applied to SRAM memory. For the experiment, four sample images are prepared; images 1 and 2 have higher contrast, texture, and frequency than image 3 and image 4. As shown in the figure, images 1 and 2 result in higher MSSIM indices than images 3 and 4 even though a same error rate is applied to all images.

As Cho *et al.* presented the analysis of SRAM memory behavior under aggressive voltage scaling in [87], some changes in the operation of a cell do not cause an error in the operations of other cells based on the parallel architecture of memory. In addition, the error rate is determined without respect to the input values of the memory

Table 1: Summary of cases that cause long delay propagation length in terms of the sign and the magnitude of addends

A / B	Negative Large	Negative Small	Positive Small	Positive Large
Negative Large	_	*Case 3* $P_{df} \propto \lvert A-B \rvert$	_	*Case 2* $P_{df} \propto 1/\lvert A+B \rvert$
Negative Small	*Case 3* $P_{df} \propto \lvert A-B \rvert$	_	*Case 1* $P_{df} \propto 1/\lvert A+B \rvert$	_
Positive Small	_	*Case 1* $P_{df} \propto 1/\lvert A+B \rvert$	_	*Case 3* $P_{df} \propto \lvert A-B \rvert$
Positive Large	*Case 2* $P_{df} \propto 1/\lvert A+B \rvert$	_	*Case 3* $P_{df} \propto \lvert A-B \rvert$	_

* *A, B*: addends * P_{df}: Probability of delay faults,
* $a \propto b$: *a* is proportional to *b* * sign(*a*) : sign bit of *a*
Case 1: sign(*A*) ≠ sign(*B*) & magnitude of *A* and *B* is small
Case 2: sign(*A*) ≠ sign(*B*) & magnitude of *A* and *B* is large and similar
Case 3: sign(*A*) = sign(*B*) & magnitude difference of *A* and *B* is very large

cell. Therefore, for the analysis of the relationship between the voltage level of memory and the quality of the output image, the natural disparity in error tolerance is the main consideration. In contrast to memory, the architecture of logic includes several paths, which are consisted of serially connected logic components. Any change in a logic component such as value and delay can affect subsequent components. For these reasons, under aggressive voltage scaling, logic requires delay-oriented fault analysis in addition to the natural disparity in error tolerance.

4.3 Arithmetic error and image quality

The frequency of the input combinations that cause a long delay propagation path is proportional to the probability of delay faults. According to the analysis in Chapter 3.2.2, the number of consecutive full adders with *01* or *10* input determines the length of a delay propagation path. This section categorizes the input combinations that cause a long delay propagation path into three cases, explains all the cases with examples, and summarizes them in Table 1.

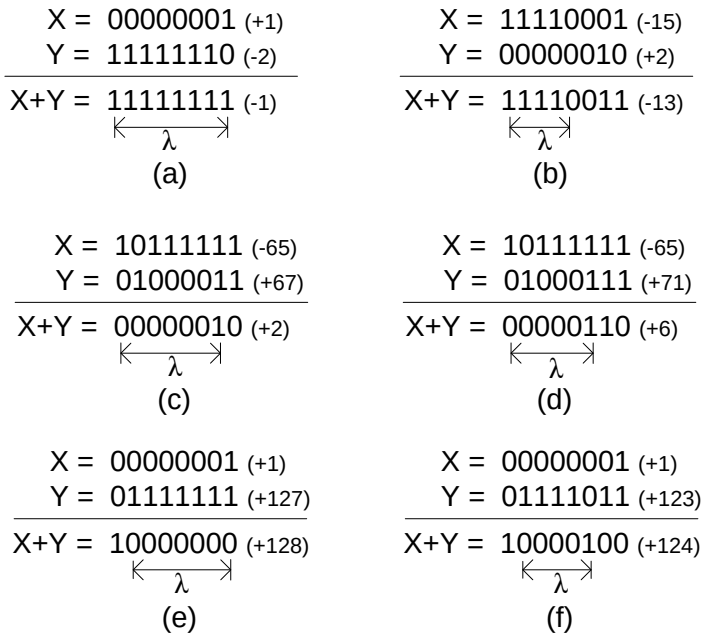


Figure 4.3: Illustrative examples using 8-bit adder (λ: delay propagation length) for *case 1*: (a), (b), *case 2*: (c), (d), *case 3*: (e), (f).

First of all, if one addend has mostly 0s and the other has mostly 1s, then it satisfies the condition of long consecutive *01* or *10* input. This example is *case 1*, in which the signs of two addends are different and the magnitudes of the both numbers are small. *Case 1*, which also includes the case in which one of addends is zero, is illustrated in Figure 4.3 (a). The addition of two numbers *x* and *y*, *x = 1* and *y = -2*, has long consecutive *01* inputs. In this case, the difference in the magnitude of the two numbers is inversely proportional to the delay propagation length (e.g., see the difference between Figure 4.3 (a) and (b)). The other case (*case 2*) is the addition of two different signed numbers in which the magnitudes of the two numbers are large and similar. The two close numbers have a similar bit pattern, and the complement of an addend results in an opposite bit pattern from the other. Figure 4.3 (c) illustrates this case. The addition of the two numbers *x* and *y*, *x = -67* and *y = 66*, results in long consecutive *10* or *10*. In this case, the difference in the magnitude of the two numbers is inversely proportional to the delay propagation length (e.g., see the difference between Figure 4.3 (c) and (d)). The last

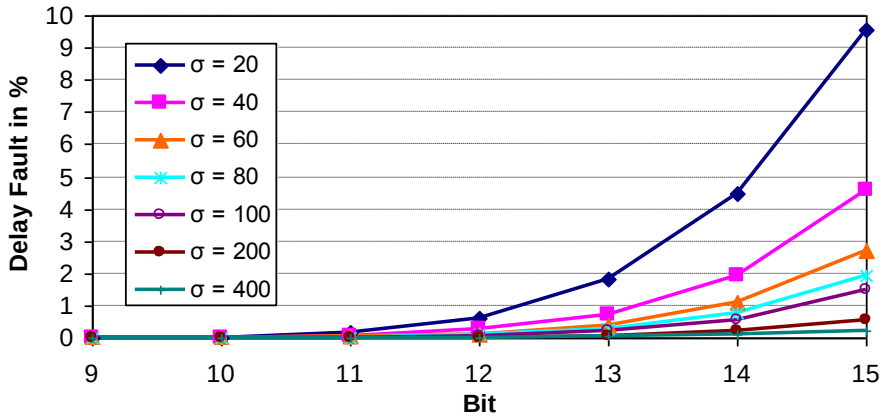


Figure 4.4: Delay fault rate comparison for various standard deviations of Gaussian random inputs ($\kappa = 0.72$, results for 0~8bit is 0).

case (*case 3*) is the addition of two same signed numbers in which the magnitude difference between addends is very large. As shown in Figure 4.3 (e), the addition of two numbers *x* and *y*, *x = 1* and *y = 127*, causes a long delay propagation path. Difference in the magnitude of two numbers is proportional to the length of the delay propagation path in this case (e.g., see the difference between Figure 4.3 (e) and (f)).

To verify how the magnitude of addends affects the delay fault rate, we simply compare the delay fault rate for a 16-bit ripple carry adder using Gaussian random numbers (100,000 input sets) with a zero mean and various standard deviations. For Gaussian random numbers, the numbers with large magnitude are very rare, especially when the standard deviation is small. Thus, *case 1* and *case 2* are dominant. Since our simulation is based on two's complement representation, the sign extensions of the addends for *case 1* and *case 2* frequently propagate the generated delays up to the MSB. As a result, the MSB has the highest probability of a delay fault. As shown in Figure 4.4, the probability of a delay fault was increased for the inputs with a smaller standard deviation which raises the occurrence of *case 1* inputs. Simulations with different voltage scaling factors followed the same trend.

It is important to note that the final delay value at each output bit is frequently not large enough to cause a delay fault even though an input combination leads to a long

propagation path because the relationship between a previous and a current operation determines the delay value of each component on the path. In other words, the delay fault rate can be reduced by manipulating the order of inputs. He *et al.* proposed a technique to reduce the length of delay propagation path by grouping operand with the same sign [88]. Thus, only the same signed numbers are accumulated and the results of two accumulations are added at the last. This technique targets the accumulation of large number of data since it requires significant hardware overhead for additional accumulator and control units. Therefore, in this paper, we do not include any analysis of an input order, nor apply techniques for manipulating the order.

4.4 Empirical analysis of voltage scalability dependent on image characteristic

This section shows that different input images lead to very different output quality under aggressive voltage scaling using various test images and conditions for JPEG encoding application. Based on the fact that the voltage level is determined by the output quality, it discusses the difference in the energy saving is possible if the input image type is considered during voltage scaling.

4.4.1 Experimental setup

DCT in JPEG

The 2-D DCT is the most computationally intensive process among encoding processes. Our analysis and simulation is based on the algorithm presented in [52]. It is assumed that the DCT algorithm is implemented using the pipelined architecture described in [89, 90], in which each stage includes only one arithmetic unit. Figure 4.5 illustrates the architecture. Excluding the multiplication stages, a total of 10 adders are required for the 2-D DCT. It is assumed that all addition stages use 16-bit ripple carry adders. The simulation tool for accurate error analysis discussed in Chapter 3.4 is

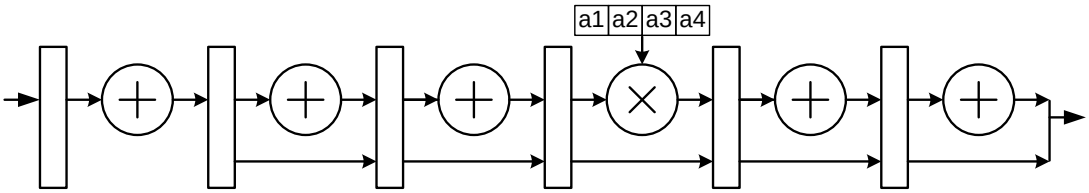


Figure 4.5: Illustration of a pipelined 1-D DCT implementation. Each stage performs a maximum of eight addition operations. a1, a2, a3, and a4 are coefficients for constant multiplications.

integrated into the JPEG program included in MediaBench [91]. The modified program performs delay computations for all addition processes in the 2-D DCT and then generates erroneous values based on the computed delay results under a specified voltage. With this simulation setup, the analysis of the dependence of output quality on input images can be performed easily by observing the error rate for each adder and the quality of output images.

Simulation setup for process variation

Under aggressive voltage scaling, the impact of process variation such as threshold voltage (V_t) fluctuation on the propagation delay of a circuit is not negligible since the relationship between V_{dd} and V_t is directly related to the delay as explained in Chapter 2. In order to observe variations in delay faults and error rates due to process variation, a random V_t shift is applied for each transistor in the full adder. Two components of process variations are considered: (a) D2D variations, in which the amount in threshold voltage shift (ΔV_t) is the same for all transistors in a die, and (b) WID variations, in which each transistor in the die experiences different ΔV_t. It is assumed that ΔV_t follows a Gaussian distribution with a zero mean and a standard deviation of 30mV for both D2D and WID process variations. For an n-bit ripple carry adder, one global V_t variation is chosen first for all n fulladders, and then local V_t variations are prepared for each transistor in each fulladder. With the V_t assignments,

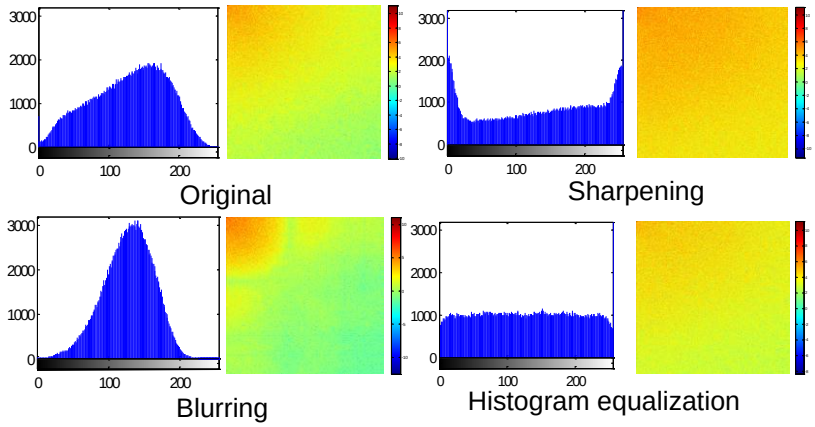


Figure 4.6: Example histograms and logarithms of DCT magnitudes. Red color means larger value (energy) and blue is the opposite. For blurred images, large values are concentrated in low frequency region. In contrast, for sharpened or histogram equalized images, the values are spread into all regions.

delay values for 64 different inputs are obtained for each fulladder. For one global V_t assignment, we consider 100 different local V_t assignments. This complete process is repeated for all considered supply voltage assignments, and the entire process is repeated for 100 different global V_t assignments.

4.4.2 Results and discussion

Relationship between image types and voltage scalabilities

To analyze the dependencies of error rates on input images under scaled voltages, several different image types are prepared by using three common image processing techniques that excite or diminish the conditions that cause delay faults: blurring, sharpening, and histogram equalization. Blurring attenuates high-frequency contents; in contrast, the sharpening and the histogram equalization boost high-frequency contents. As shown in the differences among the example histograms and among the images in the frequency domain (see Figure 4.6), these image processing techniques change the input distribution and the frequency response.

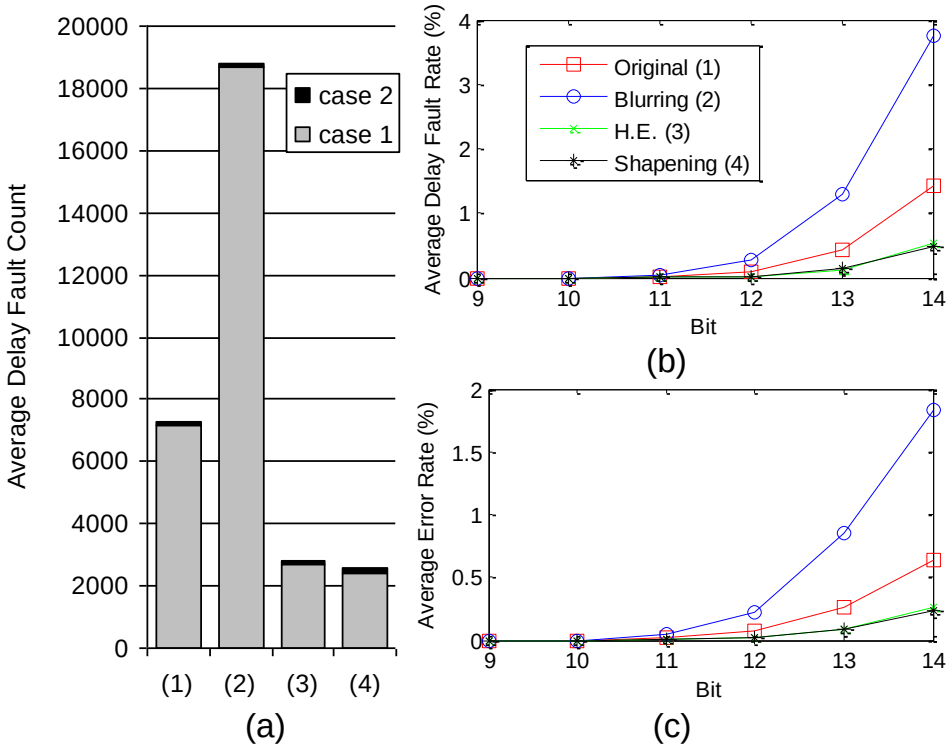


Figure 4.7: (a) Average delay fault counts for case 1 and case 2 (κ = 0.67), (b) Average delay fault rate, (c) Average error rate (κ = 0.67, results for 0~9 bits are zeros).

The additions in DCT algorithm are based on butterfly architectures (i.e., basic butterfly operations can be expressed as $y0 = x0 + x1$, $y1 = x0 - x1$). For such operations, if neighboring pixel values are close, the subtraction of the pixel values result in output values with small magnitudes. In the next stages, the addition or the subtraction of these small magnitude numbers may increase the probability of a long delay propagation path. Figure 4.7 (a) shows average delay fault counts for each input condition. *Case 3* was very rare, so it is not included in the plot. In this experiment, for *case 1*, the magnitude of addends is smaller than 32. For *case 2*, the magnitude of addends is larger than 32, but the difference in the magnitude between two addends is less than 32. As shown in the figure, *case 1* was dominant for all different test results compared to the others. The blurring process causes neighboring pixels to become close to each other by assigning the average value of neighboring pixel values to each pixel. On the other hand, the contrast

enhancing processes such as sharpening and histogram equalization increase the number of high-frequency components. As a result, the differences between neighboring pixel values become larger. In other words, the process lowers the probability of having small magnitude values or close values. The average rates of delay faults and errors for the adders in 2-D DCT are shown in Figure 4.7 (b) (c). The error rate is always less than the delay fault rate since the captured output may be correct at the presence of a delay fault. The blurred images caused a significant increase in delay fault rates and error rates while the sharpened and histogram equalized images reduced them. Note that this characteristic is consistent with the natural disparity of error tolerance discussed in previous section.

The errors generated at each adder may propagate to the next stages or be concealed during the quantization process in JPEG encoding. Although this research does not include any detailed analysis of error propagation, it is obvious that higher error rates for each adder will result in more degradation in the quality of final outputs. Figure 4.8 compares the average MSSIM index of the 50 test images (see Appendix B) for each image type [92-94], and Figure 4.9 shows the average energy consumption with respect to voltage levels. The results demonstrate that the quality of output images under aggressive voltage scaling highly depends on input image types. For the voltage level greater than the factor of 0.83, the output images did not include any erroneous pixels since operating delay is set to 1.15 T_{max} (the maximum measured delay). Below the factor of 0.83, the contrast-enhanced images resulted in higher average MSSIM indices than that for original images. In contrast, the blurred images resulted in very sharp decline in the MSSIM index as the supply voltage is scaled. For example, based on the average MSSIM index for the voltage scaling factor of 0.76, the sharpened images resulted in about 22% less quality degradation compared to the original images while the blurred images resulted in about 40% more quality degradation. This results show that voltage scalability varies dependent on input image types. Figure 4.10 illustrates the visual difference in the quality of the output images. Since the variation in energy savings for

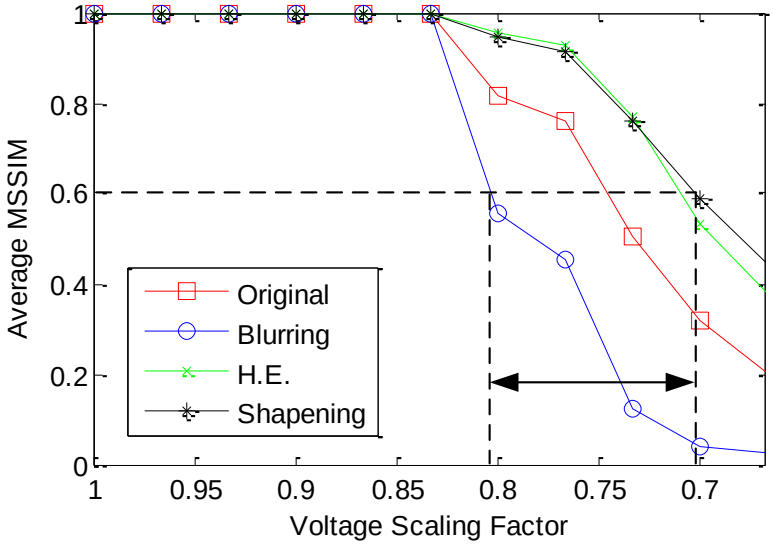


Figure 4.8: Comparison of output image qualities (Average MSSIM) for different image types.

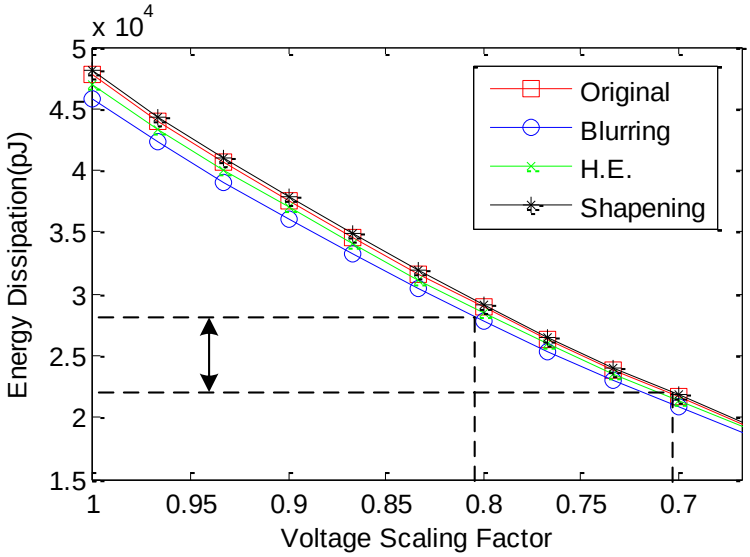
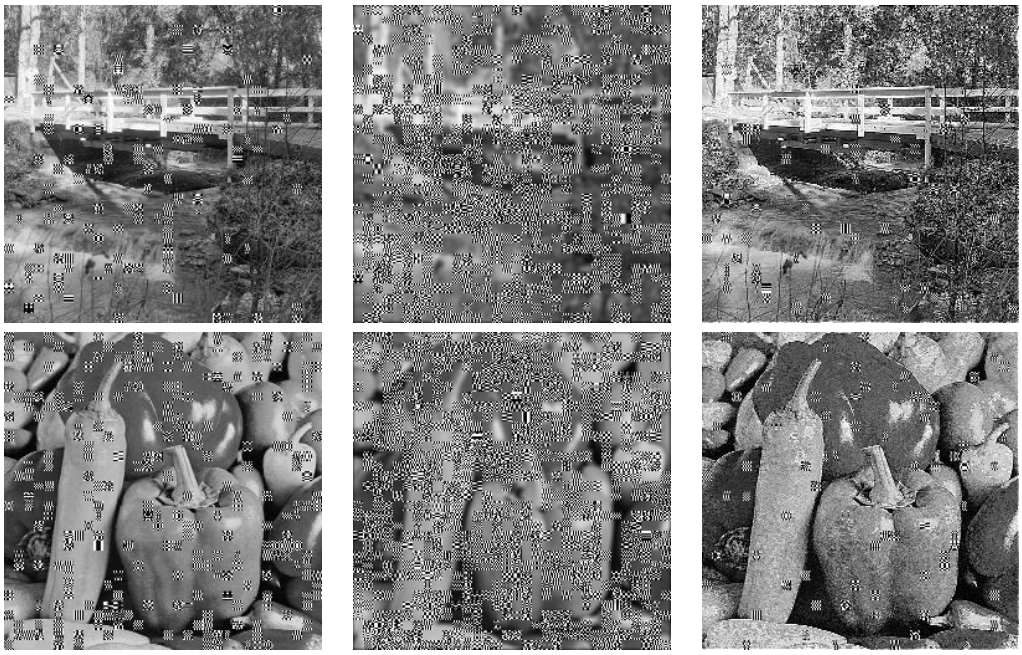


Figure 4.9: Average energy dissipation for different image types.

the different image types is trivial (less than 3%, see Figure 4.9), energy savings are primarily dependent on the voltage scalability for a certain output quality requirement. For instance, if the MSSIM index of 0.6 is the quality requirement, energy savings can vary by about 17% between two different image types.



(a) (b) (c)

Figure 4.10: Example output image quality comparisons (κ = 0.73) (a) Original (b) Blurring (c) Sharpening.

Impact of process variation

This subsection discusses the effects of process variations on the relationship between image types and energy savings. The two main concerns are i) whether the differences in voltage scalability among the different image types are valid under process variations, and ii) how large the variations in the difference are. Figure 4.11 (a) shows the results of the average MSSIM index under process variations for each image type while the supply voltage is scaled. Compared to the results without process variations, the image quality degradation began earlier and the voltage range for the degradation was wider because the error rate under process variations is determined by both the supply voltage level and the threshold voltage fluctuation. The average voltage difference between blurred images and sharpened images ($\Delta V_{b\text{-}s}$) at an MSSIM index of 0.6 is about 120mV, and the corresponding energy savings is about 17%. As shown in Figure 4.11 (b), $\Delta V_{b\text{-}s}$ gradually decreases as the MSSIM index requirement increases. Figure 4.11 (c) shows the histogram of $\Delta V_{b\text{-}s}$ for the MSSIM index of 0.5. If the $\Delta V_{b\text{-}s}$ result is a

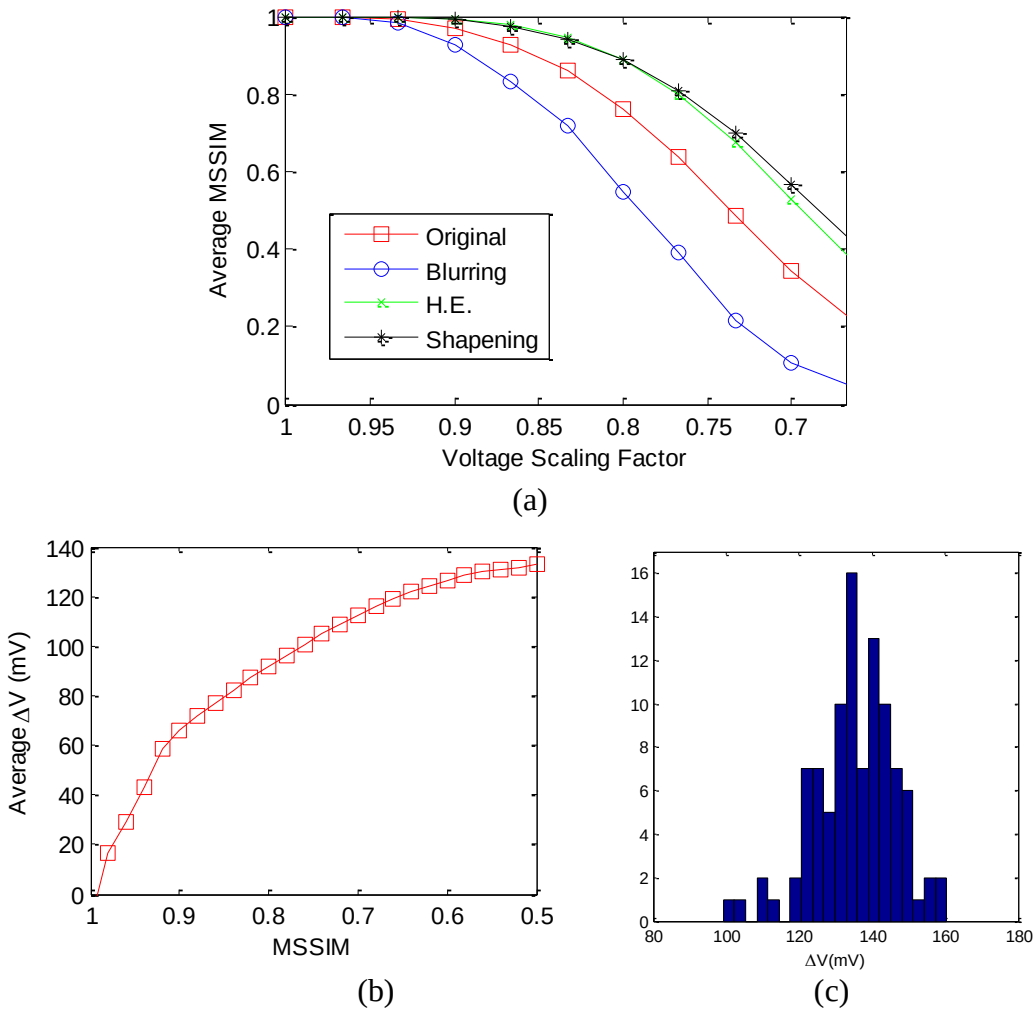
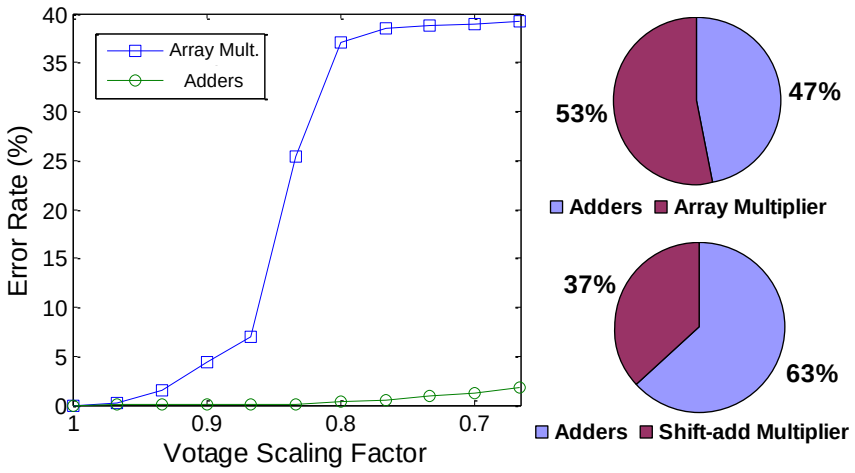


Figure 4.11: (a) Comparison of output image qualities (Average MSSIM) for different image types under process variation, (b) Average $\Delta V_{b\text{-}s}$ (voltage difference between sharpened images and blurred images), (c) Histogram of $\Delta V_{b\text{-}s}$ for MSSIM of 0.5.

Gaussian distribution, the standard deviation (σ) of $\Delta V_{b\text{-}s}$ is in the range of 11~15mV, which is trivial compared to the mean of $\Delta V_{b\text{-}s}$. The 3-σ worst-case value of $\Delta V_{b\text{-}s}$ is about 100mV, and the 3-σ best case value is about 170mV. These results show that differences in voltage scalability among the image types are still valid under process variations.



(a) (b)

Figure 4.12: (a) Error rate comparison, (b) Average energy consumption comparison.

Impact of aggressive voltage scaling on multipliers

As explained in Chapter 4.4.1, the DCT process requires a constant multiplication stage. This subsection explains the impacts of aggressive voltage scaling on two different multipliers. In the first, we consider a modified 14×14 array multiplier, which does integer multiplication and provides shifted output, instead of the floating point multiplication. Since one operand is always one of four different fixed constants [52], we optimized the operating frequency for the multiplier so that it results in zero timing slack, which is the same for the adders. As we discussed in Chapter 3.4.2, the error rate of the multiplier is much higher than that of the adders. Figure 4.12 (a) shows the error rate results for both the multiplier and the ripple carry adder. We may not apply voltage scaling to the multiplier because the sharp increase in the error rate results in severe quality degradation. Although a significant amount of energy is devoted to the multiplier (see Figure 4.12 (b)), for a given quality requirement, the case that applies voltage scaling to only the adders results in the most efficient quality-energy tradeoff among the three cases, as shown in Figure 4.13.

The second is a constant shift-add multiplier. For many DSP applications such as DCT and filters, when the one operand of multiplication is always constant, shift-add

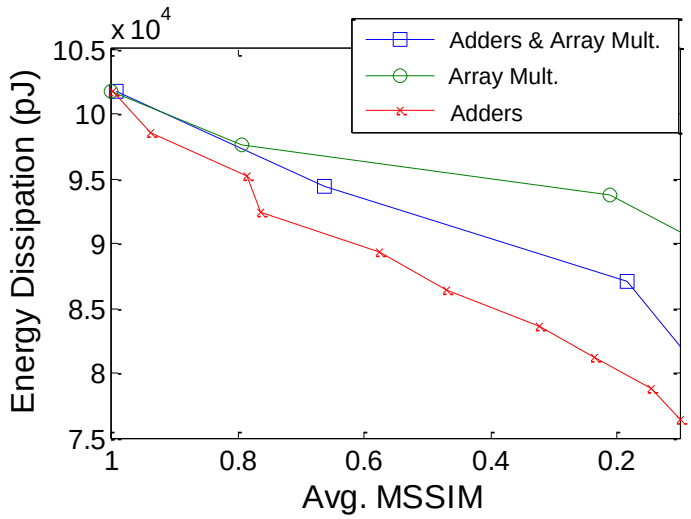


Figure 4.13: Energy dissipation with respect to image quality degradation.

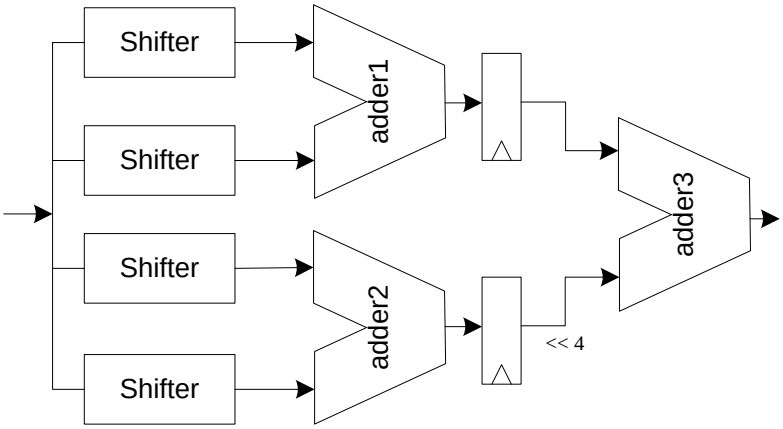


Figure 4.14: Shift-add multiplier structure.

multipliers are often used since it consumes less energy from less hardware requirement. We consider the pipelined shift-add multiplier explained in [90], which requires three adders and shifters. Figure 4.14 shows the architecture of the multiplier. For adder1 and adder 2, the values of an operand are always the shifted value of the input. The operands for all adders are always the same signed numbers. The only input patterns are (+N')+(+N'') and (−N')+(−N''), where N is the non-constant operand and N' and N'' are shifted values of N from the shifters. Therefore, based on the analysis in Chapter 3, almost all input sequences to the three adders do not result in a long delay propagation

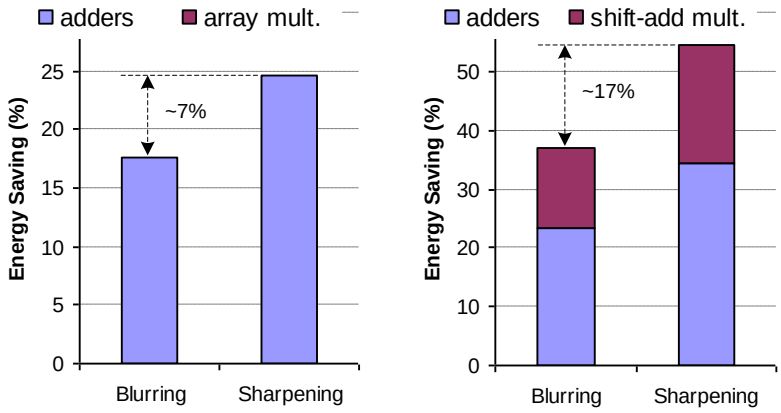


Figure 4.15: Comparison of difference in average energy saving at MSSIM=0.6 between two different image types (left chart: no energy saving for array multiplier).

path. The pattern does not cause long consecutive *10* or *10* inputs, and the shifted bits do not require additions. The experimental results show that errors from adders were dominant in total error rate for all considered voltage levels. The output image quality results are almost identical to Figure 3.7, which only includes adders. Figure 4.15 shows an example comparison of difference in energy savings between different image types with respect to two different multipliers.

Impact of technology node and image size

This subsection explains the impact of changes in i) the technology node and ii) the image size on the difference in voltage scalability between image types. For these experiments, process variation is not considered, so the operating delay is set as T_{max}, which results in zero timing slack. Three technology nodes, 70nm, 45nm, and 32nm, are selected. Nominal supply voltages for the corresponding technology nodes are 1.2V, 1.0V, and 0.9V. It is obvious that the difference between the supply voltage and the threshold voltage (V_{dd} - V_t) becomes smaller as scaling technology node. In addition, with the same voltage scaling factor, the percentage change in (V_{dd} - V_t) for a smaller technology node is greater than that for a larger technology node. For these reasons, as

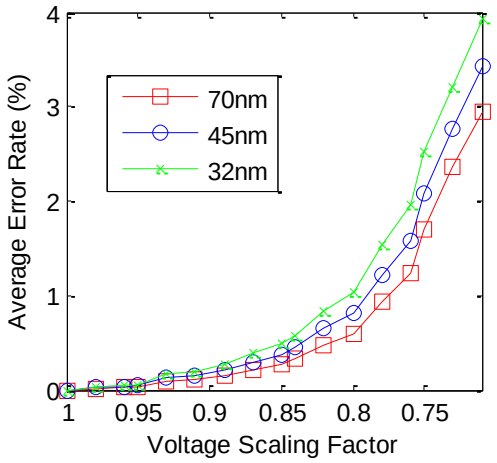
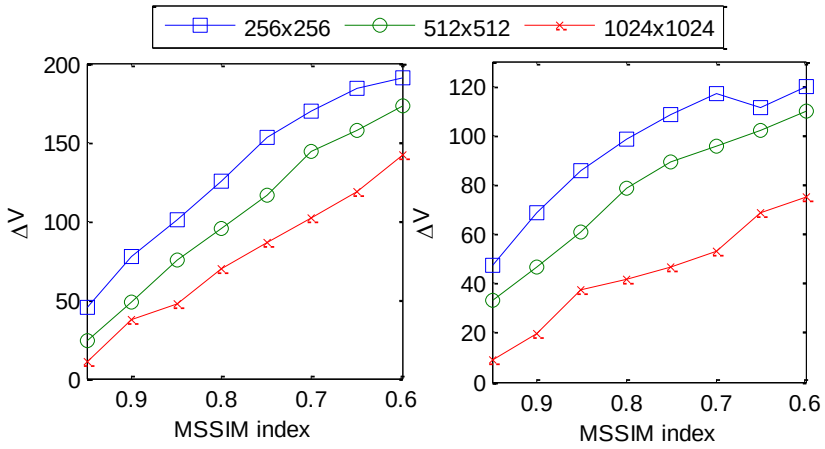


Figure 4.16: Average error rate of MSB for different technology node.

the supply voltage is scaled, the propagation delay increases more quickly for smaller technology nodes. This behavior results in a higher error rate for the smaller technology node, as shown in Figure 4.16.

In addition to the technology node, the size of an image affects the voltage scalability. The DCT for JPEG performs 8x8 blockwise operations. Generally, an 8x8 block of a larger image includes less visual information compared to an 8x8 block of a smaller image. In other words, in an 8x8 block, the probability that neighboring pixels will have similar values increase as image size increases. Therefore, compared to the same small images, larger images reduce the difference in voltage scalability. For our simulations, three different sizes of images are prepared by resizing 1024x1024 images to 512x512 and 256x256 images. After applying same image processing techniques such as blurring, sharpening, and histogram equalization to all the test images, we observed average changes in the voltage scalability between the two different image types with respect to the size difference (see Figure 4.17 (a)). Figure 4.17 (b) shows the comparison results for two different original images that exhibit a noticeable difference in voltage scalability. The results validate that the voltage scalability decreases as image size increases.



(a) (b)

Figure 4.17: Comparison of voltage scalability (a) between sharpened images and blurred images, (b) between two different images.

Impact of spurious transition on energy estimation

The delay estimation in this dissertation is based on a zero-delay model, which ignores spurious transitions [95]. A real-delay model accounts for all possible transitions including glitches, which requires significantly increased complexity and simulation time for a behavioral simulation. The main purpose of the delay estimation is to check the presence of delay error. Therefore, a zero-delay model is sufficient if the delay estimation follows the delay propagation path from low-order bits to high-order bits since the maximum input delay is added to the output delay for each component. However, energy estimation based on a zero-delay model may not provide correct results since energy consumption due to spurious transitions is not negligible. To show the impact of spurious transitions in energy estimation, NanoSim is used to count all possible switching activities using real input vectors for considered test images. A 16-bit ripple carry adder is designed in Verilog and synthesized using OSU 180nm technology at a nominal voltage. All input data to the adders in DCT is captured and prepared as inputs for the simulation. First five standard test images are used for this simulation (see Appendix B). Figure 4.18 shows average spurious switching activity count compared to total switching activity count. Total switching activity counts are almost similar for different types of

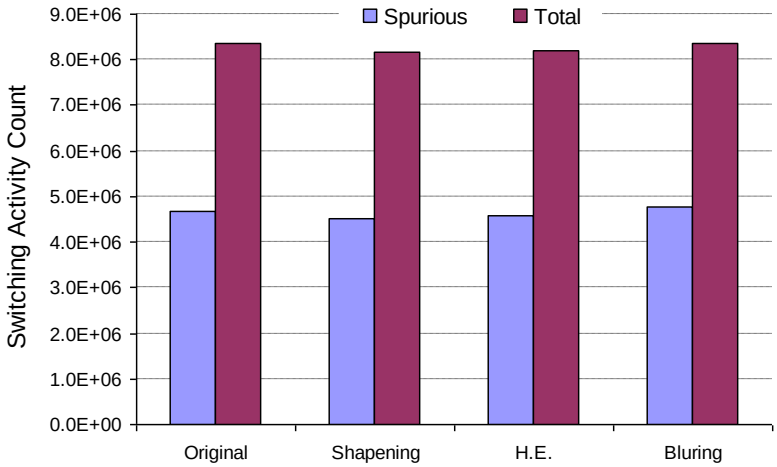


Figure 4.18: Comparison of switching activity count for the different types of images.

images, and the spurious transitions occupy almost 50% of total transitions. Sharpened and histogram equalized images resulted in a little smaller number of total and spurious transition, while blurred images resulted in small increase. Note that this characteristic is consistent with the analysis of the relationship between input image type and energy savings. However, the impact on spurious transition is very small compared to that on voltage scalability.

4.5 Summary

This chapter presents an analysis of the dependence of energy consumption on input images when aggressive voltage scaling is applied to error tolerant DSP applications. The experimental results using JPEG encoding show that voltage scalability based on output quality varies for input image types. An interesting observation is that images with higher contrast, strong texture, and high-frequency information are more error tolerant than those with opposite characteristics in terms of two aspects, the natural disparity in error tolerance due to the characteristics of the human visual system and the different rate of error dependent on input images under aggressive voltage scaling. For a given output image quality requirement, more error-tolerant input images can be operated at lower voltages than others. By assigning an optimal supply voltage based on the image

type, energy dissipation can be minimized. We believe that the energy savings resulting from the relationship between input images and output quality will have great implications for low-energy image processing system design.

CHAPTER 5

SYSTEM-LEVEL ENERGY ANALYSIS AND OPTIMIZATION

5.1 Introduction

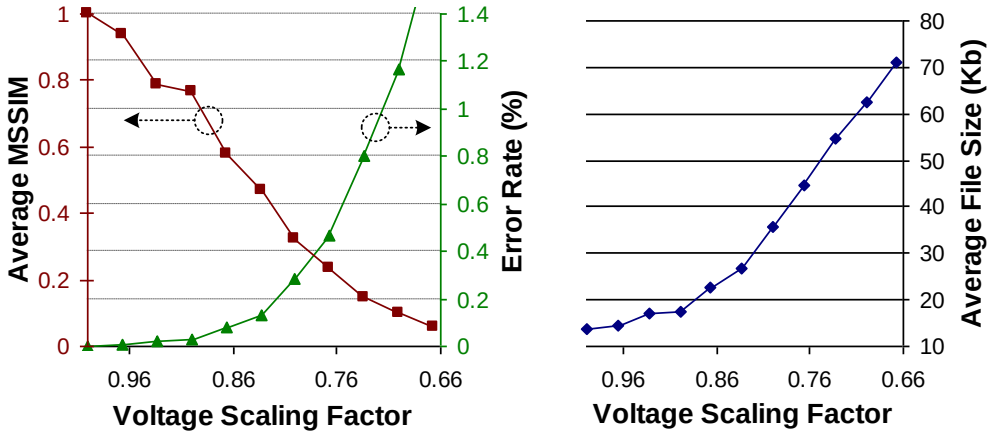
A fundamental shortcoming of the existing efforts using aggressive voltage scaling for accuracy-energy tradeoffs in image/multimedia applications is the lack of analysis of its effect on overall quality of solutions – not only the quality of output image/video. From a system-level point of view, overall energy saving may not be always obtained by trading off the output image quality under aggressive voltage scaling. In this chapter, we first present a fundamental analysis of aggressive voltage scaling on overall system energy. Aggressive voltage scaling to the combinational logic blocks in the discrete cosine transform not only degrades the quality of the output image, but it also reduces the compression ratio, which increases file size of compressed image. Increased file size may result in more energy consumption in other subsystems, such as memory for storing the output images. This increase in energy consumption for other subsystem may cancel out the energy savings from aggressive voltage scaling. Thus, system-level analysis is necessary for determining usefulness of the aggressive voltage scaling technique.

Based on the analysis, we present an adaptive bit truncation method to achieve optimal tradeoffs between overall application quality (image quality and compression ratio) and system energy. The proposed method is based on the concept of pixel truncation which has been proposed earlier for low-power video compression applications [67]. Pixel truncation has been considered as a factor to trade off the accuracy of computation for switching activity reduction. By truncating less significant bits, it minimizes the impact of inaccurate computations while reducing switching energy

consumption. This chapter discusses that how the technique can be utilized to effectively reduce the error rate due to aggressive voltage scaling with very small implementation cost. The error rate reduction directly improves voltage scalability by suppressing overall output quality degradation and the increase in file size. In terms of energy consumption, it is expected to have benefits from both of voltage scaling and switching activity reduction. Therefore, with adaptively adjusting the number of truncation bits, the system can be optimized with respect to both energy consumption and output image quality.

5.2 System level energy analysis for image compression

As discussed before, the increase in delay due to voltage scaling is the main source of erroneous operations that cause quality degradation. The increase in delay for a prior component may affect the error occurrence of the later components in propagating paths for arithmetic units. In other words, the delay for the component at the end of a propagating path is affected by the delay increase in all the components in the path. Thus, in general, high-order bits are more vulnerable to delay errors since they are located at the end of critical or subcritical paths. Therefore, most erroneous operations cause a very large magnitude of error. In addition, for the multiple stages of computation, a generated error may grow after being propagated through other stages. Figure 5.1 (a) shows the average error rate of MSB of adders in DCT and corresponding image quality degradation under aggressive voltage scaling. Due to the characteristics of error explained above, the rate of output image quality degradation is steep. Thus, noticeable energy savings requires significant image quality degradation. In addition, the reduction in the compression ratio is another significant issue of the aggressive voltage scaling techniques for image compression systems. As shown in Figure 5.1 (b), the exponential increase in the error rate results in an excessive increase in output file size. For the JPEG encoding, two reasons for the reduction in the compression ratio are discussed below.



(a) (b)

Figure 5.1: (a) Error rate and image quality with respect to voltage levels (b) Average file size increase due to voltage scaling.

- The human visual system is not sensitive to rapidly-varying brightness. During encoding processes, the quantization step reduces the amount of high frequency information that is relatively redundant to the human visual system. Thus, most of high frequency coefficients become zero or very small number after quantization process. These zeros are encoded very efficiently by exploiting the run of zeros with a zigzag order as shown in Figure A.1 of Appendix A [63]. The quantization step simply divides each component in frequency domain by a constant (quantization coefficient, shown in Figure A.2 of Appendix A), and then rounds it to the nearest integer. If the magnitude of erroneous value is greater than that of the corresponding quantization coefficient, the error cannot be concealed during quantization process. Since most of errors are large magnitude numbers, erroneous coefficients replace the zero values and reduce the average run-length of zero. The increase in non-zero values requires additional labels in encoding process, which means additional code, thus it reduces the compression ratio significantly.

- In addition, based on the Huffman table for JPEG encoding, large magnitude numbers are assigned to longer codes since they are supposed to have a lower probability of occurrence under normal operation with the nominal voltage. The erroneous DCT

coefficients with a large magnitude number, even after quantization, increase the occurrence probability of the long code. This directly increases the average codeword length, which results in the reduction in the compression ratio.

Even though image quality degradation is acceptable from a visual quality perspective, the increase in the file size is another factor limiting voltage scalability because it causes additional energy dissipation in other subsystems such as memory for storing image data. Energy overhead for the subsystem may exceed the energy savings from low voltage usage. This analysis shows that an effective method to handle the issues of aggressive voltage scaling is necessary. It also shows the importance of system-level observation and optimization in the event of a trade off between output quality and energy consumption.

5.3 Low cost error reduction technique

For fixed-point systems, the bit-width is closely related to the hardware complexity and the precision of output. Numerous previous techniques attempted to use reduced bit-width instead of full bit-width to reduce hardware requirement, which results in the reduction in circuit area and energy consumption. However, the simple reduced precision approaches do not allow flexibility at runtime. Therefore, it always results in low quality outputs. This section explains an adaptive pixel and coefficient truncation method that achieves great energy savings by effectively reducing the error rate due to aggressive voltage scaling. It also discusses the implementation of the method in memory buffers for additional energy saving as well as its impacts on image quality and compression ratio.

5.3.1 Quality degradation due to pixel truncation

An 8-bit image can be considered the composition of eight 1-bit planes [96]. High-order bit planes include visually significant information. In contrast, low-order bit

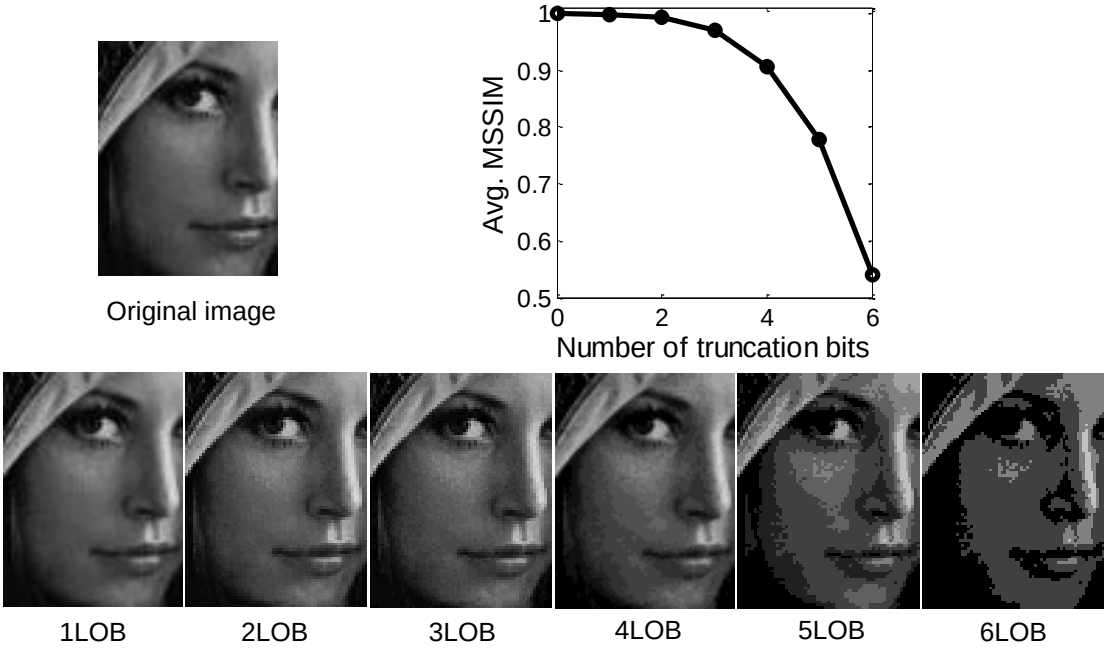


Figure 5.2: Average MSSIM degradation results and example result images.

planes include subtle intensity details. Pixel truncation, also called "bit plane slicing," reduces intensity resolution by simply turning the low-order bits (LOBs) of pixels to zeros. As shown in Figure 5.2, the average MSSIM index decreases exponentially as the number of low-order bits to be truncated increases. However, when up to three LOBs are truncated, it is not easy to sense the quality degradation with the naked eye. From the truncation of four LOBs, false contouring–a series of discrete steps instead of a continuous gradient–and rough edges start to appear in the image.

5.3.2 Error rate reduction

Because the increase in delay is the main source of erroneous operations, any methods that decrease delay may reduce error occurrences. In arithmetic computations, we can simply reduce the length of critical paths by truncating LOBs. For an example of a ripple carry adder, if three LOBs are truncated, the calculation of the critical path delay does not include the three fulladders for the truncated bits since the fourth fulladder from the least significant bit does not need to wait for the carry signal, which is zero. This

approach reduces errors due to aggressive voltage scaling very effectively since it directly diminishes the source of error. For the proposed method, the number of bits to be truncated does not change every DCT block but remains constant at least for processing an entire image. As a result, the truncated bits are zero, and they do not have any transition activity most of the time, directly reducing the propagation delay of the paths that include the bits.

The pipelined 1-D DCT architecture considered in this chapter has five addition stages and one multiplication stages (see Figure 4.5). We do not need to truncate the selected bits for every addition stages. The truncated bits of input image data remain as zeros for the consecutive addition stages. However, after multiplying cosine values, the truncated LOBs of coefficients are no longer zeros. Thus, the same LOBs of coefficients after the first 1-D DCT are truncated. Figure 5.3 (a) shows the error rate results for the truncation of the different number of LOBs. 0LOB indicates the case without any truncation technique. The result shows that error rate decreases as the number of bits to truncate increases. The comparison of the results (see Figure 5.3 (a) and (b)) shows that the additional coefficient truncation results in significant error rate reduction. Figure 5.4 shows the output quality degradation for the two cases. The additional coefficient truncation allows maintaining output images at a high quality level for much lower voltage compared to the case with only pixel truncation. Even though the pixel and coefficient truncation technique slightly degrades image quality in a high voltage range, it allows much larger voltage scalability than blind aggressive voltage scaling. Figure 5.5 shows the results of energy savings for DCT with respect to the energy consumption for the nominal case (nominal voltage and no truncation). With the proposed technique, we can expect energy savings from both voltage scaling and switching activity reduction. For the truncated bits, inputs and output are always zero, which means no switching activity. As shown in the figure, at voltage scaling factor of 0.8, it can achieve about 50% energy savings with thee bit truncation.

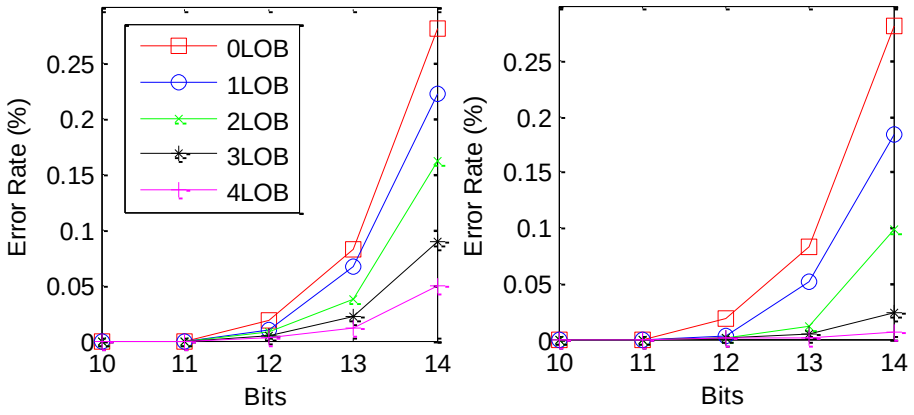


Figure 5.3: Comparison of error rate at the voltage scaling factor of 0.8, (a) pixel truncation only, (b) pixel and coefficient truncation.

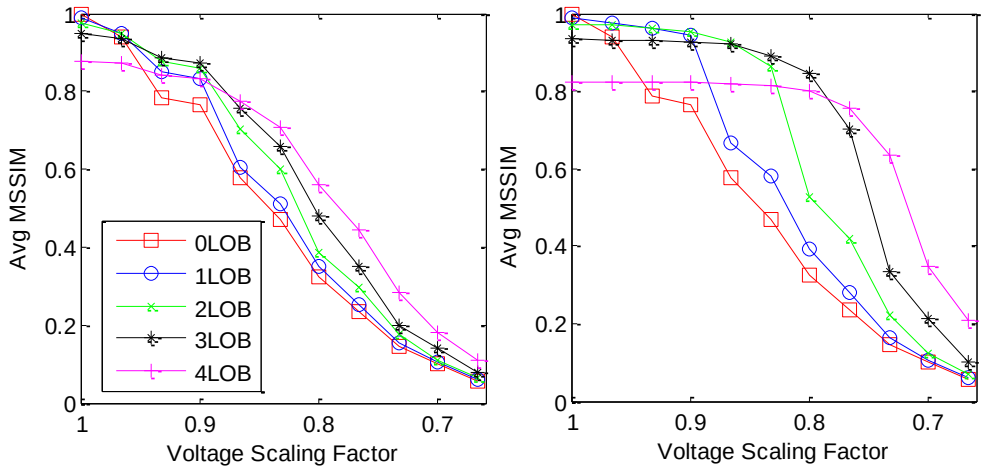


Figure 5.4: Comparison of image quality, (a) pixel truncation only, (b) pixel and coefficient truncation.

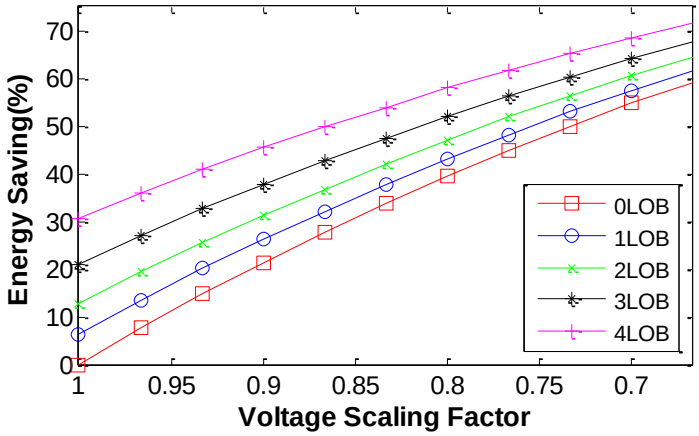
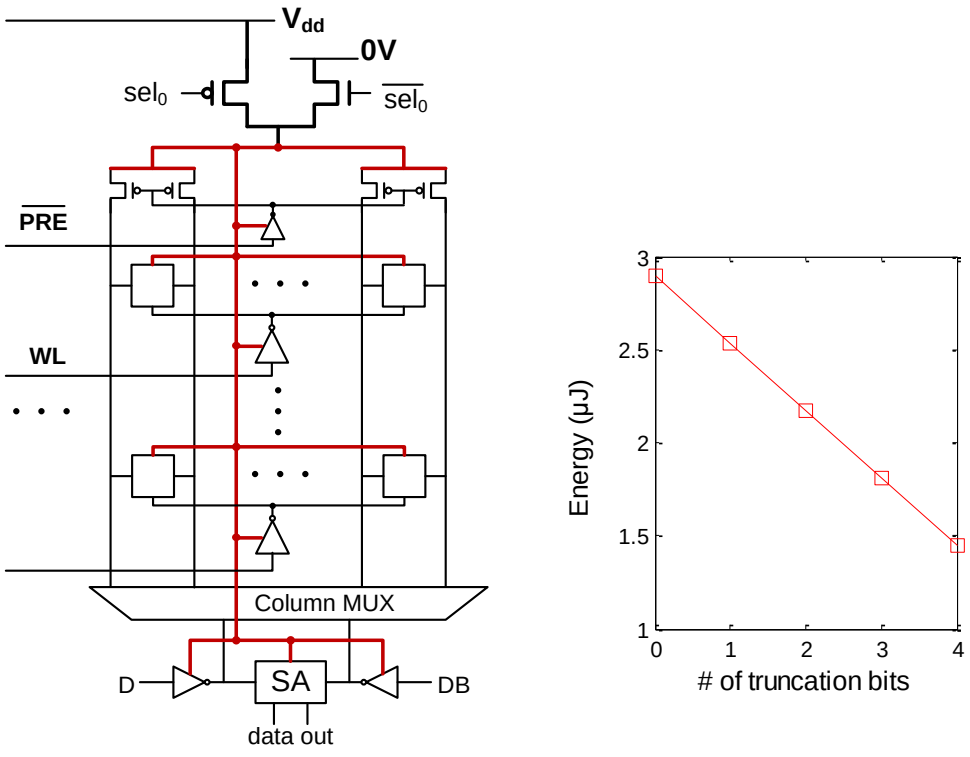


Figure 5.5: Energy saving results for DCT.

5.3.3 Implementation and circuit analysis

Generally, the truncation technique can be implemented by inserting a simple masking units on the input. Instead, the bit truncation is done in the input memory buffer and the transpose buffer since they do not need to hold the data that will be truncated at later stages. By disabling the memory cells for the bits to be truncated, the memory buffers may dissipate significantly reduced energy.

We assume that adaptive truncation is allowable up to four LOBs since the truncation of more than 4 bits results in noticeable quality degradation such as false contouring and rough edges as mentioned in previous section. Thus, we need additional 4-bit signal to gate the power source of the memory cells. The reconfigurable memory architecture proposed by Cho et al. [87] is adapted to avert possible effects on other cells during adaptive control. However, instead of lowering voltages for low-order bits as presented in [87], the voltage levels for LOBs is always 0V. This simplifies the implementation of the memory buffer compared to [87] as no second voltage source is required. Further, as the low voltage is 0V, one PMOS and NMOS device is used in the voltage reconfiguration network as shown in Figure 5.6 (a). The NMOS device is used to pass the 0V while PMOS device is used to pass the regular V_{dd}. Even though the values of the bitlines for the cells are zero, the sense amplifiers for the bitlines may provide non-zero value due to its analog property. In addition, the sense amplifiers for the cells do not need to switch likewise the memory cells for the truncation bits. Therefore, the power source of the sense amplifier is also gated. We assume that input memory buffer is SRAM and the size of memory is 4k×8, which is suitable for the test images. For transpose buffer, two 64-word 12-bit wide SRAM is used as proposed in [90]. As shown in Figure 5.6 (b), the proposed implementation can save up to about 50% energy consumption of the input buffer with only 6% area overhead for the reconfigurable architecture [87]. Since the buffers occupy small area in the system, this overhead causes a very small increase in total area.



(a) (b)

Figure 5.6: (a) Proposed architecture of input memory buffer, (b) Energy consumption for memory buffer.

5.3.4 Subsystem - DRAM

Compression ratio

Image compression such as JPEG utilizes spatial redundancy of image data. After converting data to frequency domain, quantization step reduces the amount of information in the high frequency components since the human visual system is less sensitive to the high frequency components than the low frequency. With the same quantization table, the bit truncation technique results in a little larger file size than one without the technique since it causes different characteristics of DCT output, which may require modified quantization coefficient matrix. With the given quantization table, more information is remained after quantization step. Figure 5.7 shows the comparison of the file sizes for different number of truncation bits. For the case with voltage scaling factor of 1 (error free case), file size increases exponentially as increasing the number of

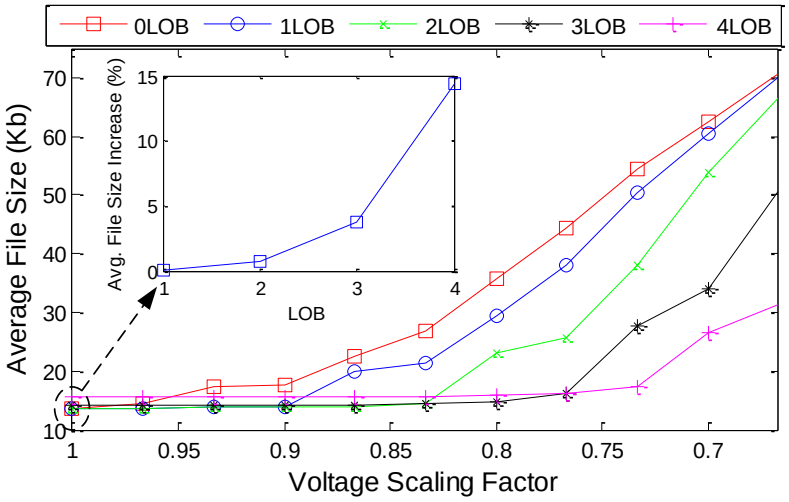


Figure 5.7: Comparison of average file size increases.

truncated bits. However, the increase is not very significant compared to the one due to aggressive voltage scaling. Instead, the pixel and coefficient truncation technique suppresses the increase in file size successfully since the technique can achieve significant error reduction.

Energy consumption

This subsection explains the analysis of energy consumption on DRAM that stores the compressed image data. It is obvious that the variation of the data size is directly related to the energy consumption of writing the data to the DRAM. Based on a power estimation result for Micron mobile DRAM explained in [97], average energy consumption for writing a compressed image file to DRAM is calculated using the equations below.

$$E_{total} = \frac{(P_{background} + P_{Activate} + P_{write})}{Bandwidth} \times Avg.FileSize \qquad (5.1)$$

The DRAM power estimation result varies widely on different DRAM configuration and technology. For this comparative study, the estimation is based on an example configuration without power management mode shown in the technical note. It is

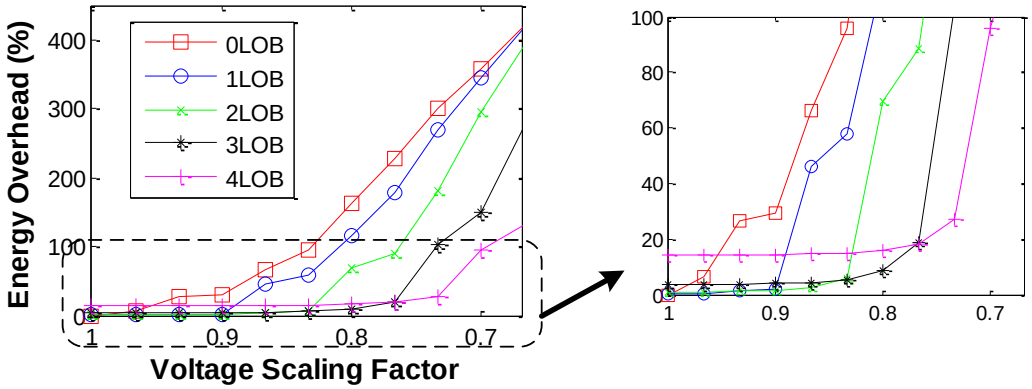


Figure 5.8: DRAM energy overhead.

assumed that the data bus is utilized only with write operations with the same total bus utilization in the example. As shown in the simulation results (see Figure 5.8), the increase in file size causes an extensive energy overhead for writing the compressed image data to the memory. Without the truncation technique, energy consumption increases right after the voltage scaling, and the overhead reaches 100% with about 200mV scaling. In contrast, the pixel and coefficient truncation technique suppresses the increase in DRAM energy consumption and maintains the low energy overhead until a much lower voltage level.

5.3.5 Adaptive control

The proposed reconfigurable architecture allows flexibility at runtime. Under various conditions such as different system constraints and operating environments, adaptive control may result in effective quality-energy tradeoffs. A simple method for adaptive control is a lookup table approach. Based on a quality requirement or a battery level, the optimal assignments of the voltage level and the number of truncated bits can be prepared as a lookup table after extensive experiments using various configurations and test images. A more complex approach includes a feedback based dynamic control. Basically, the approach requires monitoring output quality at runtime. For image/video processing systems, most of image quality metrics requires significant computational

complexity. Instead, we may use a simple linear regression model for estimating quality degradation due to voltage scaling and the truncation technique described by the equation below.

$$Q_{trunc} = \alpha_0 + \alpha_1 V_{dd} + \alpha_2 V_{dd}^2 + \varepsilon_1 \tag{5.2}$$

$$Q_{err} = \begin{cases} 0, & V_{dd} > V_{err} \\ \beta_0 + \beta_1 V_{dd} + \beta_2 V_{dd}^2 + \varepsilon_2, & otherwise \end{cases} \tag{5.3}$$

$$Q_{totol} = Q_{trunc} + Q_{err} \tag{5.4}$$

The total quality degradation (Q_{total}) is the sum of quality degradation due to truncations (Q_{trunc}) and erroneous operations (Q_{err}). This is based on the assumption that both Q_{trunc} and Q_{err} have a second order linear relationship with supply voltage. The coefficients (α, β, ε) are variables, which can be altered based on operating conditions at run time. Note that the truncation technique suppresses erroneous operations so Q_{err} is zero until a certain point (V_{dd}=V_{err}).

5.4 System-level optimization

For the comparison of overall energy consumptions, the sum of energy consumption for three components is considered: input memory buffer, DCT, and memory storage for output images. It is assumed that variations in energy consumption for other components are not significant compared to the three components. Figure 5.9 shows the result of comparisons of three cases: i) blind aggressive voltage scaling, ii) aggressive voltage scaling with 4-bit fixed truncation (for maximum energy savings), and iii) aggressive voltage scaling with adaptive bit truncation (for minimum image quality degradation). The ii) and iii) are two boundaries of the result of the truncation technique. As shown in Figure 5.9, blind voltage scaling does not really reduce overall energy consumption. It even causes the increase in overall energy consumption as the supply voltage is scaled. This is mainly because of very fast quality degradation with respect to voltage scaling as shown in Figure 5.10. The increase in energy overhead for the memory

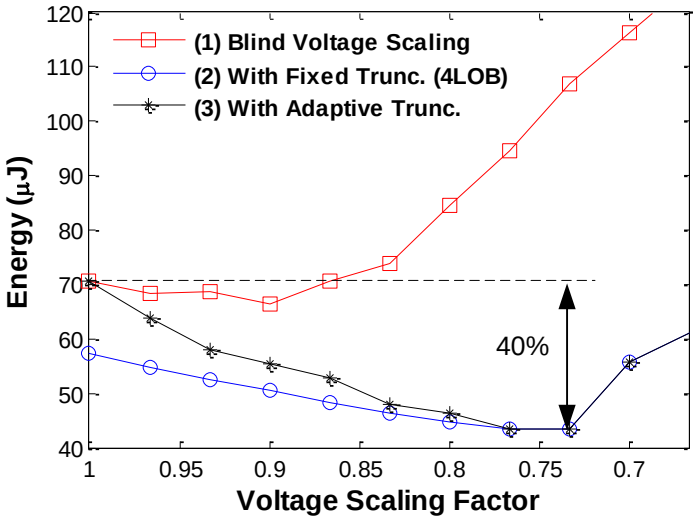


Figure 5.9: Overall energy consumption comparison.

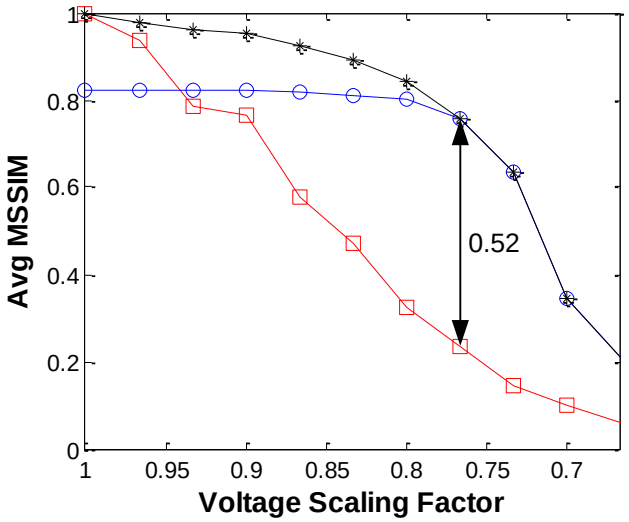
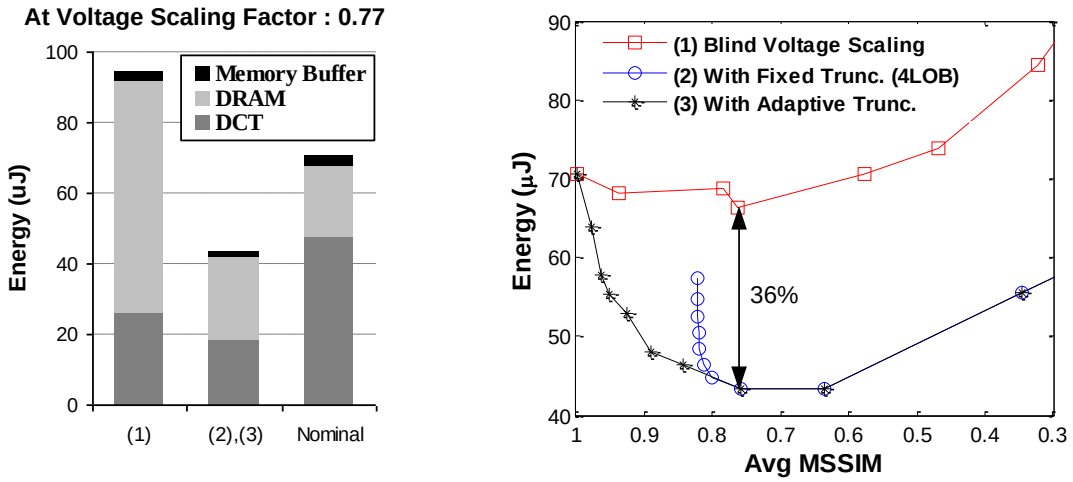


Figure 5.10: Output image quality comparison.

cancels out the energy savings in other units and overwhelms the overall energy consumption from a certain voltage level. In contrast, the proposed technique results in maximum 40% reduction in energy consumption compared to the nominal case, and it results in 0.52 higher MSSIM index compared to blind aggressive voltage scaling at the voltage scaling factor of 0.77. Figure 5.11 (a) illustrates the energy consumption of each component at the voltage scaling factor of 0.77. The figure clearly shows that the extensively increased energy consumption for the memory cause a large overall energy consumption. In contrast, the proposed approach suppresses the energy overhead for



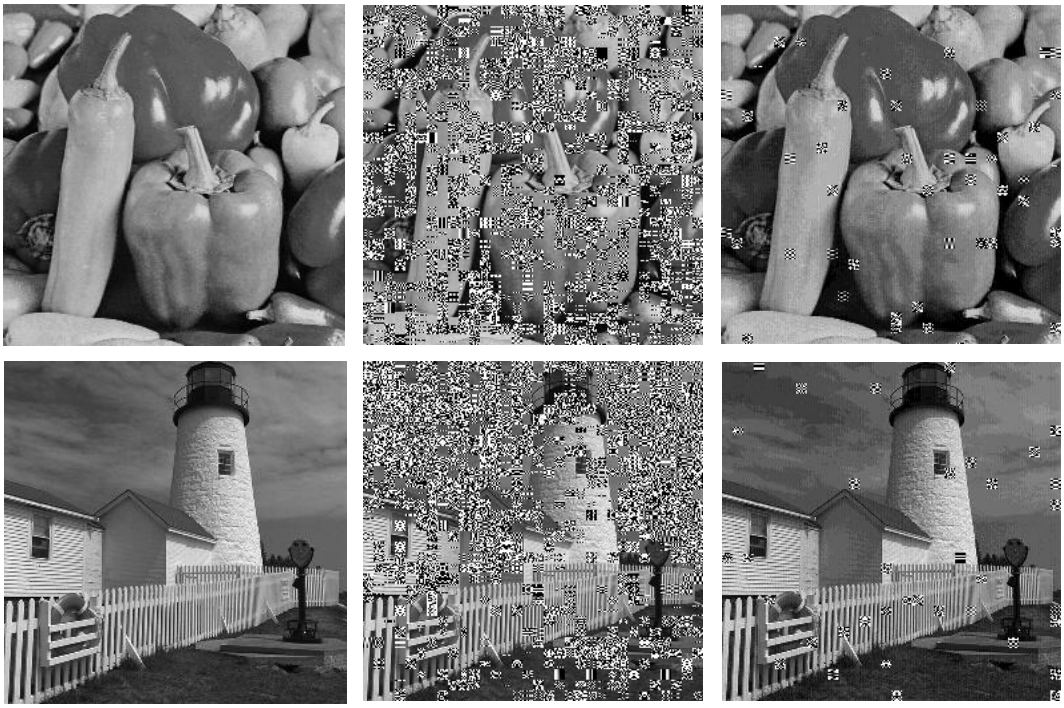
(a) (b)

Figure 5.11: (a) Energy consumption of each component, (b) Energy consumption with respect to image quality degradation.

memory and significantly reduces energy consumption for other components. As shown in Figure 11 (b), the proposed technique achieves most energy savings at high quality level (the MSSIM index of 0.8~1). Given a quality requirement (the MSSIM index of 0.76), the proposed technique achieves 36% more energy savings than blind voltage scaling. Example output images show the quality difference of the two in Figure 5.12.

5.5 Summary

This chapter discusses the impact of aggressive voltage scaling on image compression system. We demonstrate that significant reduction in compression ratio of JPEG encoding may result in extensive energy overhead in other systems such as memory for storing the compressed image data. Based on the analysis of error characteristics and its impact on system energy consumption, we present the adaptive pixel and coefficient truncation method for effectively minimizing energy consumption while maintaining high image quality under aggressive voltage scaling. Our experimental results show that the approach can successfully achieve great energy saving by adaptively change the number of truncation bits with trivial implementation cost.



(a) (b) (c)

Figure 5.12: Example comparison of visual image quality at the voltage scaling factor of 0.77. (a) original image, (b) blind voltage scaling, (c) voltage scaling with proposed technique (4-bit truncation).

CHAPTER 6

EFFICIENT ACCURACY-ENERGY TRADEOFF BASED ON ERROR CONCEALMENT

6.1 Introduction

Until previous chapter, we focus on the impact of delay errors on final output quality and discuss a method to reduce the error rate. The previously presented method is very simple and easy to implement with trivial hardware overhead. However, the approach affects all data computations without respect to the error occurrence. In this chapter, we present a voltage scalable JPEG encoder architecture based on an error concealment method. The goal of this approach is to trade off the accuracy of computations only at the presence of an error in order to achieve efficient accuracy-energy tradeoffs. For the two dimensional discrete cosine transform (2-D DCT) in an image compression system, a delay error can cause significant impact on output image quality because the error has mostly large magnitude and may grow during subsequent computations. As shown in Figure 6.1, it may ruin the entire 8x8 block. We first present a simple error detection technique followed by the error analysis of DCT unit. Then, we

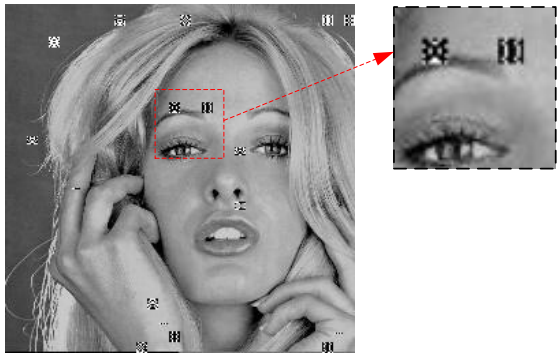


Figure 6.1: Example output image under aggressive voltage scaling.

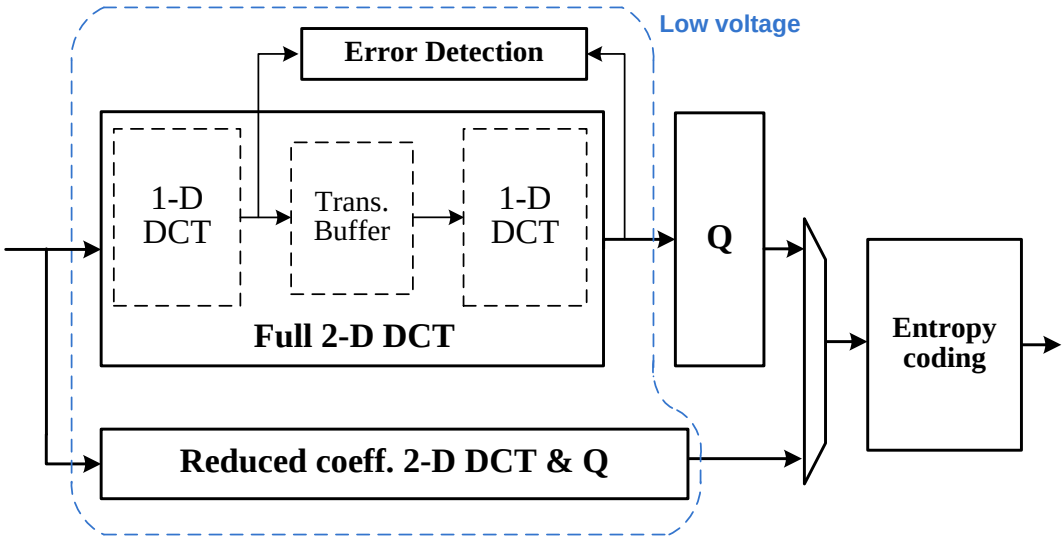


Figure 6.2: Proposed JPEG encoding architecture.

introduce an error concealing method using a reduced coefficient 2-D DCT so that we can minimize quality degradation while saving energy consumption. Based on experimental analysis, we discuss energy savings for several components in the image compression system. In addition to great energy savings in computational units under aggressive voltage scaling, the proposed technique also results in energy savings in memory for storing the compressed image data by increasing compression ratio. The experimental results show that the proposed approach can exhibit significant overall energy savings with a small hardware overhead.

6.2 Efficient accuracy-energy tradeoff based on error concealment

Figure 6.2 illustrates the proposed architecture. Additional components compared to original architecture are a reduced coefficient 2-D DCT unit (RCDCT) and a corresponding quantization unit, an error detection unit, and a control unit. The RCDCT unit provides somewhat inaccurate results but not erroneous results. As mentioned in previous chapters, in this dissertation, it is assumed that a result is erroneous when it is distorted by a timing failure. After analyzing the characteristics of erroneous DCT

outputs under aggressive voltage scaling, we introduce a simple error detection method that requires trivial implementation cost. Whenever an error is detected, by simply substituting the erroneous output with the output from the RCDCT output, we can maintain the final output images in a high quality level. The main purpose of this architecture is not only to maintain the high quality level of output but also to allow low voltage operations for the computationally intensive components as marked in Figure 6.2

6.2.1 Error analysis and simple error detection

As discussed in Chapter 5.2, for arithmetic units such as adders and multipliers, the increase in delay for a prior component may affect the error occurrence of the later components in propagating paths. In other words, high-order bits are more vulnerable to delay errors since they are located at the end of critical or subcritical paths. Therefore, the magnitude of most erroneous outputs is very large. In addition, for the multiple stages of DCT computation, an erroneous operation in early stages may affect all the related subsequent computations, and the generated error may grow after propagating through other stages. Therefore, erroneous DCT outputs have high probability to have very large magnitude values in more than one DCT coded coefficients.

Based on the characteristics of errors under aggressive voltage scaling, we can simply detect if the outputs include an erroneous value. In general, the first coefficient of 1-D DCT output (F(0), see Figure 6.3) is often large magnitude value compared to other outputs (F(1)~F(7)). In addition, the probability to have an error in F(0) is much smaller than others since it requires fewer computations compared to that for other outputs. As shown in Figure 6.3, F(0) and F(4) requires only three pipelined stages without including a multiplication while others do more than four stages with a multiplication stage. For these reasons, we examine the magnitude of seven outputs (F(1)~F(7)) for the error detection. Figure 6.4 compares the distribution of the magnitudes of correct and erroneous values. When the output bit-width is 16-bits, most of correct values are less

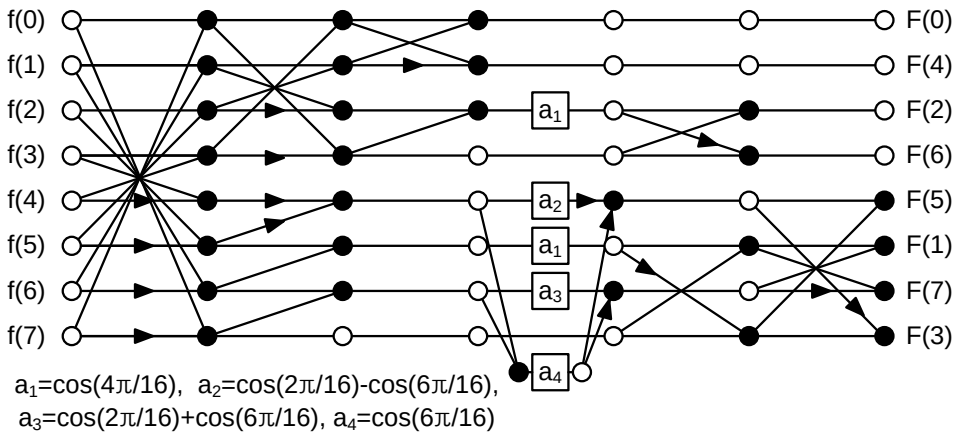


Figure 6.3: Flow graph of 1-D DCT algorithm in [44].

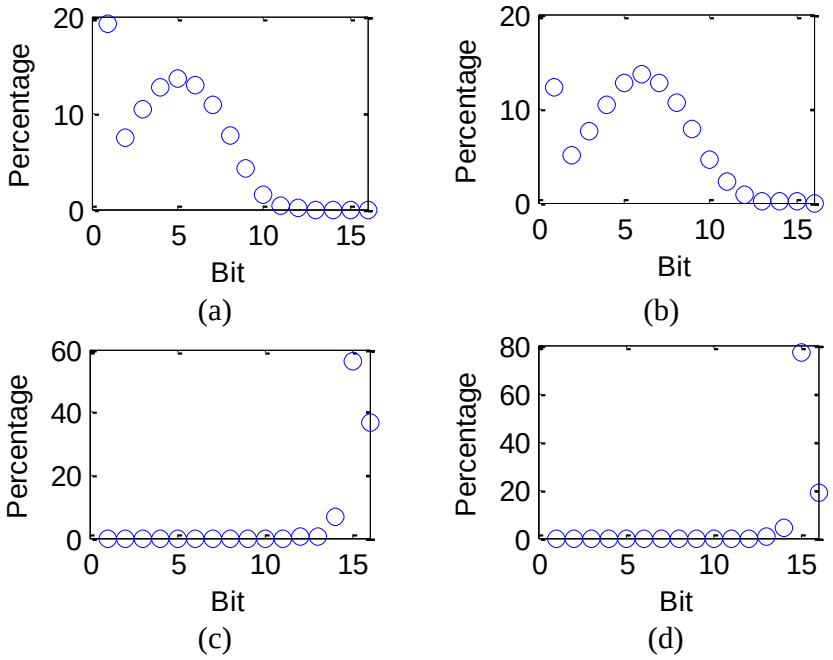


Figure 6.4: Magnitude distribution for (a) the correct values of the 1st 1-D DCT, (b) the correct values of the 2nd 1-D DCT, (c) the erroneous values of the 1st 1-D DCT (d) the erroneous values of the 2nd 1-D DCT.

than 13-bits. In contrast, erroneous DCT outputs are mostly larger than that. For the two's complement number representation, correct values have sign extension in HOBs, which are the same bit values with sign bit. In contrast, erroneous outputs have at least one bit in HOBs that is different from the sign bit. For the example of one bit error case, if the sign

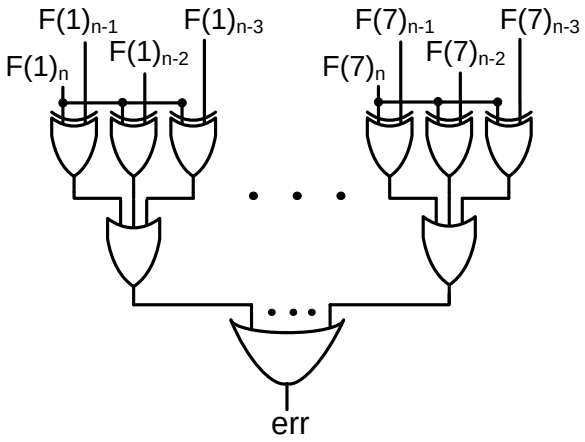


Figure 6.5: Error detector architecture. In this case, three bits, $(n\text{-}1)^{th}$, $(n\text{-}2)^{th}$, and $(n\text{-}3)^{th}$, are compared with the sign bit.

bit is erroneous, the sign bit is different from other sign extension bits. If $(n\text{-}1)^{th}$ bit is erroneous, then the bit is different from the sign bit. Therefore, by comparing the sign bit and selected HOBs as shown in Figure 6.5, we can simply detect erroneous output. This error detection technique requires only small number of gates, which is a trivial hardware overhead.

6.2.2 Reduced coefficient DCT architecture

The reduced coefficient 2-D DCT computes only DC coefficient (c_{00}) and first horizontal and vertical AC coefficients (c_{01}, and c_{10}). The signal energy of 2-D DCT output is concentrated on a few low frequency components, which are coefficients at the left upper corner. The lossy compression concept of JPEG algorithm is based on this property to achieve the high compression ratio by reducing information in high frequency components. After quantization step, most of coefficients become zeros or very small numbers. Therefore, the three coefficients are often sufficient to represent an 8x8 block of image data. The output of the RCDCT substitutes for the erroneous output from the full 2-D DCT (FDCT) unit to maintain relatively high output quality level.

Since the RCDCT is additional component to the existing image encoding system, hardware overhead is the main consideration when we design it. Although the RCDCT

calculates only three coefficients, it requires considerable computational complexity due to the correlation among DCT computations. To effectively reduce the computational complexity, we use subsampled input data to compute the three coefficients. Instead of using the full 8x8 input image block, we use the subsampled 4x4 block, as shown in Figure 6.6. The 4x4 block based RCDCT needs only ¼ of inputs compared to 8x8 block based RCDCT. To examine the impact of the subsampling on output image quality, we simply compared the difference in image quality between the 8x8 and 4x4 block based approaches for all test images. Within the considered voltage range, the differences in average MSSIM indices were less than 0.01, which means that the impact of subsampling on final output quality is trivial.

The first step of the 4x4 block based RCDCT is to calculate the sum of each row and column. Using these horizontal and vertical sums (*h_sum* and *v_sum*), we compute the three coefficients as explained in the equations below.

$$h\_sum_i = \sum_{j=0}^{3} a_{ij}, \quad v\_sum_j = \sum_{i=0}^{3} a_{ij} \tag{6.1}$$

$$c_{00} = \frac{1}{2}\sum_{i=0}^{3} h\_sum_i \tag{6.2}$$

$$c_{01} = m1 \times (v\_sum_0 - v\_sum_3) + m2 \times (v\_sum_1 - v\_sum_2) \tag{6.3}$$

$$c_{10} = m1 \times (h\_sum_0 - h\_sum_3) + m2 \times (h\_sum_1 - h\_sum_2) \tag{6.4}$$

$$\textit{where} \quad m1 = \cos(2\pi/16) + \cos(6\pi/16)$$
$$m2 = \cos(2\pi/16) - \cos(6\pi/16)$$

The proposed 4x4 block based RCDCT architecture has two advantages in terms of minimizing overhead. As we mentioned above, compared to the FDCT, the RCDCT requires significantly reduced computational complexity, which is directly related to hardware requirement. The RCDCT requires only four multiplications and 31 additions. In addition, we may ensure that this unit does not cause any delay error under very low voltage levels. In other words, the voltage scalability of the RCDCT unit allows

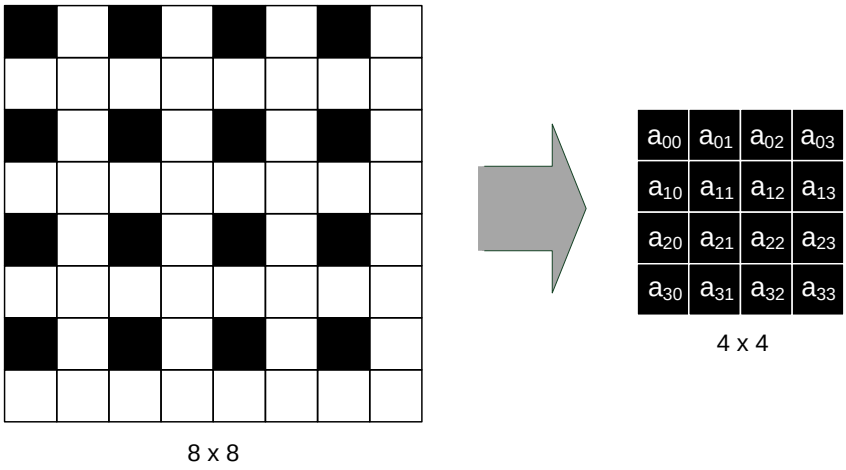


Figure 6.6: Input subsampling.

minimizing energy overhead. For the FDCT architecture, each row or column of input pixel data is inserted into the pipelined DCT unit. However, the operations of the RCDCT are based on two clock periods since the input data is inserted in every other cycle. This allows significantly reduced supply voltage for the RCDCT unit without causing a delay error.

In addition, since most of outputs from the RCDCT unit are zeros, we do not need to run the full quantization unit for the three coefficients. Instead, the quantization of the three coefficients can be done with dedicated shift and add operations. This reduction in switching activity and processing time for the quantization unit decreases energy consumption.

6.2.3 Redundant switching activity reduction

Whenever an error is detected, the remaining procedures of the FDCT are completely redundant because the RCDCT unit substitutes for the FDCT. Therefore, we gate inputs to subsequent units in the FDCT to reduce switching activity for additional energy savings. For example, when an error is detected in the first 1-D DCT unit, inputs to the transpose memory are gated so that we can reduce switching activities in the

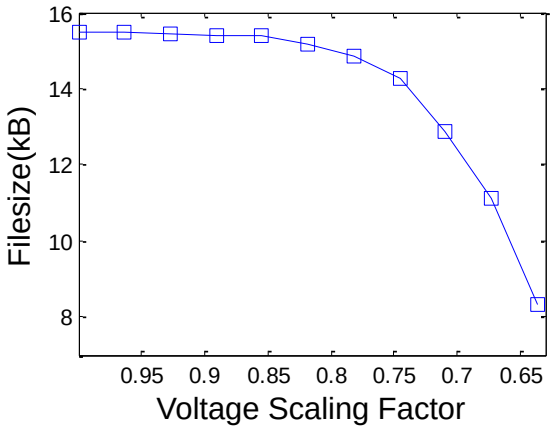


Figure 6.7: Average file size with respect to voltage.

transpose memory, the second 1-D DCT, and the quantization. When an error is detected at the outputs of the second 1-D DCT, we gated inputs to the quantization. Instead of input gating, powering down the component at redundant operations may also reduce leakage energy consumption. However, for the pipelined architecture, power gating brings several problems. First, the component may not be turned off until the previous input to the pipeline is processed completely. In addition, the frequent power switching need to address several issues such as excessive in-rush current and delay for switching [98]. For these reasons, power gating is not considered in this approach.

6.2.4 Energy reduction in memory

The proposed technique also reduces energy consumption for other subsystem such as memory to store the compressed image data. The RCDCT provides numerous zero coefficients that increase the run-length of zeros, which is directly related to compression ratio. Therefore, as the frequency of the substitution increases, the output file size decreases as shown in Figure 6.7. If we assume that coded image data is stored to memory, it is obvious that the file size reduction results in energy savings for the memory since it requires fewer write operations. The estimation of the energy consumption for memory is based on the method explained in Chapter 5.

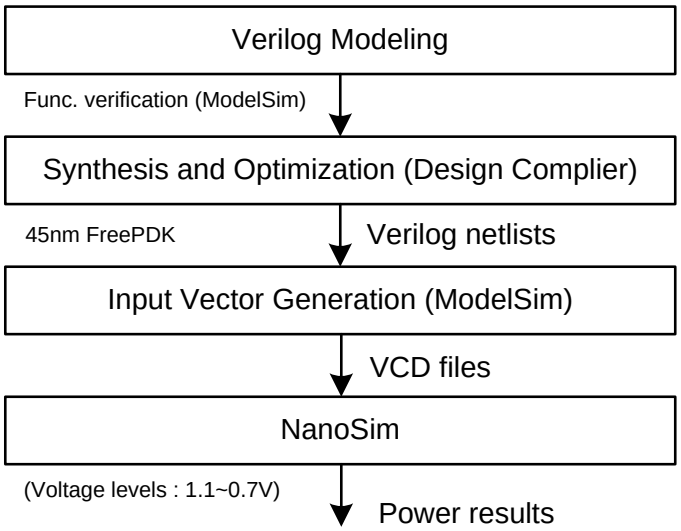


Figure 6.8: Power estimation procedure.

6.3 Experimental framework

Spice-level experiments for the error analysis are based on the experimental framework explained in Chapter 3. We used a maximum operating frequency that does not cause a delay error under process variation at the nominal voltage level (1.1V). The HSPICE simulation of the smallest component (fulladder) for all possible input transitions were based on 45nm predictive technology model. To observe the impact of process variation on the error rate and output image quality, we applied a random threshold voltage shift (ΔV_t) for each transistor in the fulladder. We assumed that ΔV_t follows a Gaussian distribution with a zero mean and a standard deviation of 30mV for both inter-die and intra-die process variations.

In order to obtain accurate energy consumption results for various voltage levels, we followed the procedure shown in Figure 6.8. We first implemented the proposed design shown in Figure 6.2 in Verilog. For the full 2-D DCT and the RCDCT units, we used 16-bit adders for all additions. The implemented design was synthesized using Synopsys Design Complier and FreePDK to obtain a Verilog netlist [83, 99]. Then, we

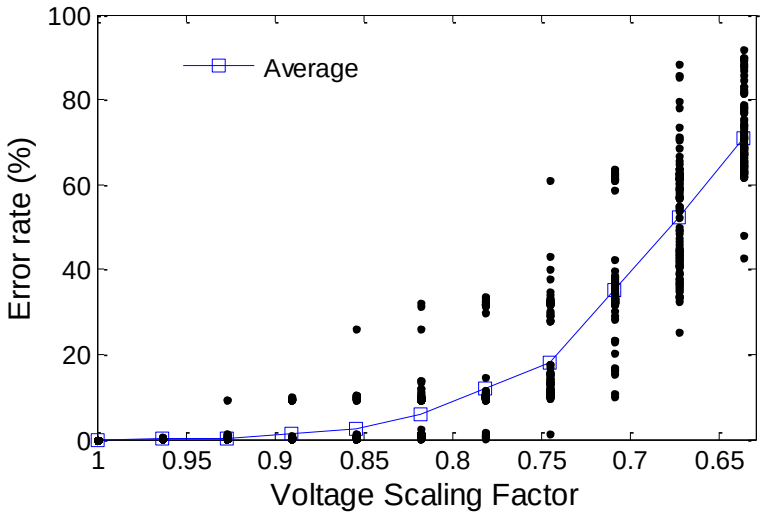


Figure 6.9: Error rate under process variation.

generated input vector using Modelsim for the first five standard test images shown in Appendix B. Using the netlist and input vectors, power consumption is measured using NanoSim for various voltage levels.

6.4 Results and discussion

Since the RCDCT generates only three DCT coefficients at the top left corner, it results in some blocking effect and blurring for the parts that need high frequency information. However, it causes very small impact on the area that does not have details. As we discussed in Chapter 4, compared to image data with numerous high frequency components, blurred image data has higher probability to cause erroneous operations. This means that more substitutions may be occurred in the regions with relatively flat image data, and this will results in less quality degradation compared to the opposite case. In addition to the location of RCDCT usage in image data, the error rate is another significant factor to determine output image quality since it is the same with the rate that RCDCT output substitutes FDCT output. Figure 6.9 shows the error rate of the FDCT unit for considered various voltage levels. Multiple errors may occur for an 8x8 DCT computation, but we only count it as one. Once an error is detected, the DCT output for

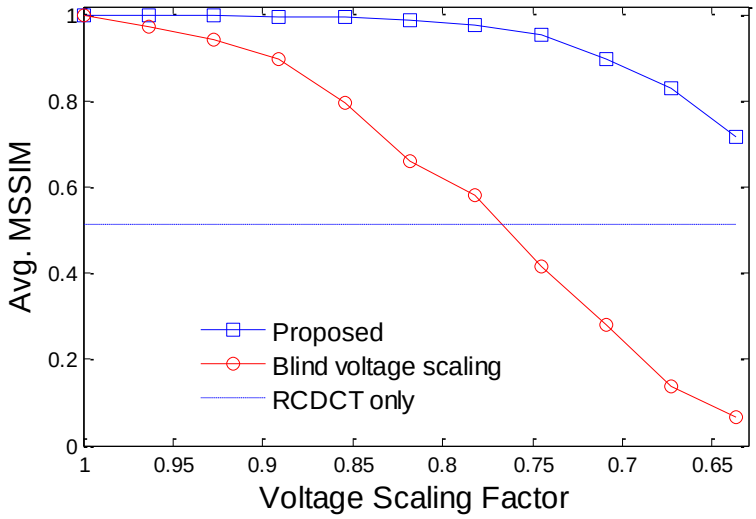
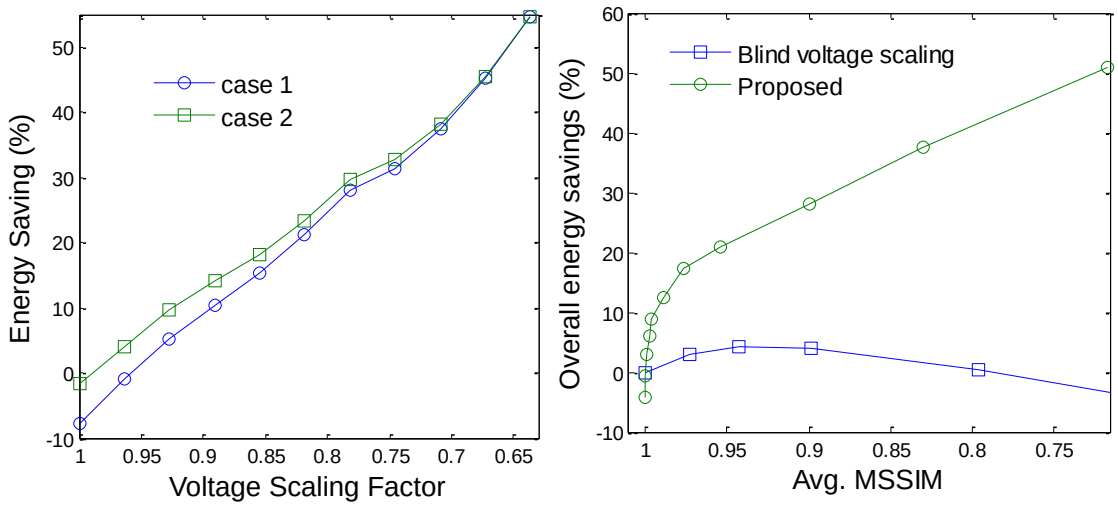


Figure 6.10: Comparison of quality degradation.

the entire 8x8 block is substituted for the output from the RCDCT, so we do not need to distinguish between multiple error cases and single error cases. As shown in the figure, the error rate increases exponentially as the supply voltage is scaled. At the voltage scaling factor of 0.63, the average error rate reaches almost 70%. This means that more number of output coefficients is from the RCDCT instead of the FDCT. One can consider an extreme case that simply replaces the FDCT with the RCDCT. In this case, we do not have a control over image quality. It always results in relatively very low image quality, the average MSSIM of 0.5145. In addition, as shown in the figure, the error rates are spread in a wide range under process variation. We can expect that worst-case design approaches that attempt to prevent any erroneous operations are not efficient to deal with such variations. In contrast, by addressing only erroneous cases, the proposed method provides efficient accuracy-energy tradeoffs.

Figure 6.10 shows the comparison of average quality degradation due to blind voltage scaling and voltage scaling with the proposed technique. While blind voltage scaling causes significant quality degradation because of errors, the proposed technique maintains high image quality until very low voltage levels. Note that the quality degradation for the proposed technique is not directly from errors. Instead, the



(a) (b)

Figure 6.11: (a) Energy savings over conventional design, (b) Overall energy savings with respect to image quality degradation.

insufficient DCT computation from the RCDCT is the direct source of quality degradation. Although the proposed technique results in energy overhead for additional hardware requirements such as the RCDCT and the error detection units, it allows achieving great energy savings from both low voltage operations and switching activity reduction. Figure 6.11 (a) shows the energy savings of the proposed architecture over a conventional design without voltage scaling. For the RCDCT, we can either apply the same voltage scaling for both the FDCT and the RCDCT (case 1) or use the lowest possible voltage for the RCDCT (case 2) to minimize the energy overhead. Case 2 is based on the assumption that more than two different power supply rails are available. As shown in the figure, at the voltage scaling factor of 0.71, the average energy saving is about 37% with the average MSSIM index of about 0.9. The area overhead for the proposed architecture is about 8.5%.

In addition, as discussed in previous section, the proposed technique also results in energy savings for the memory that stores the compressed image data. Figure 6.11 (b) shows the comparison of overall energy savings with respect to image quality degradation. This result includes the energy consumption of both the image compression

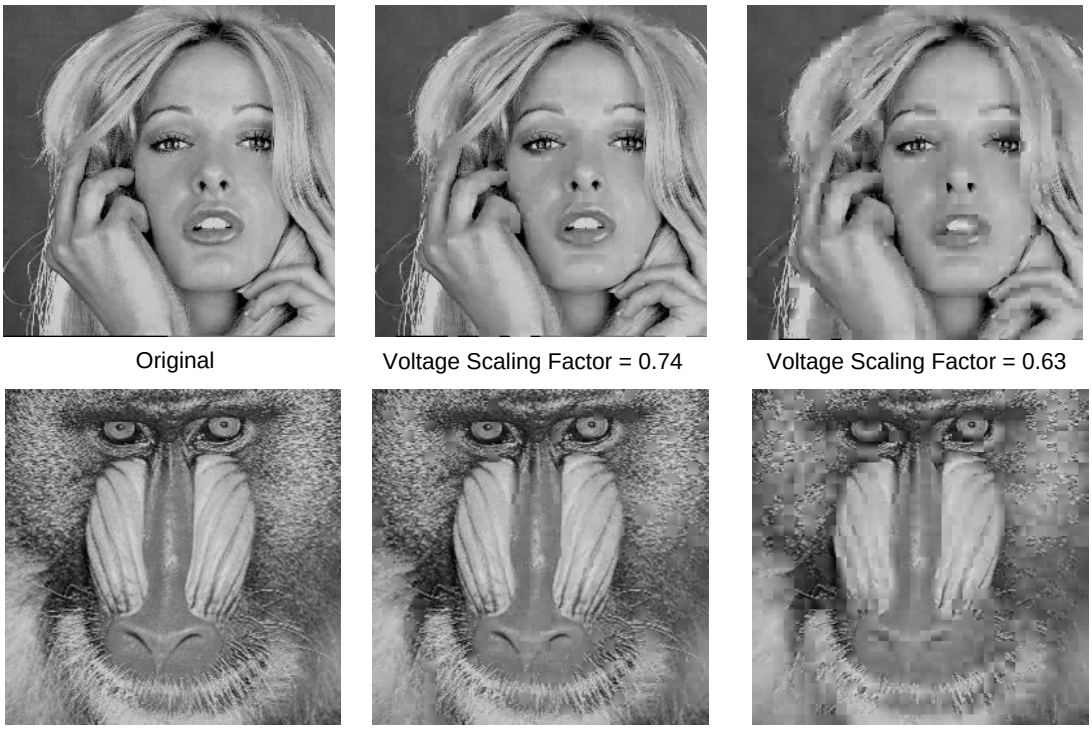


Figure 6.12: Example output images.

units and a memory. For the blind voltage scaling case, increased energy consumption due to increased file size cancels out the energy savings in computational units as we discussed in Chapter 5.3. For the reason, the blind voltage scaling case does not achieve overall energy saving even under very low voltage. However, the proposed technique results in significant overall energy savings because of the energy reduction in both computational units and a memory. Figure 6.12 shows example output images for visual comparison.

6.5 Summary

Based on efficient accuracy-energy tradeoffs, this chapter presents a low-energy image processing system design in the presence of process variation. The proposed technique conceals erroneous outputs due to delay errors under aggressive voltage scaling. With the simple error detection mechanism, the erroneous DCT outputs under scaled

voltages and process variation are substituted for the output from the reduced coefficient DCT. Using the proposed technique, we can allow significantly reduced supply voltage for great energy savings without noticeable quality degradation. Since the technique also increase the compression ratio, in addition to the computational units, the energy savings in memory to store the compressed image data improve the efficiency of the tradeoffs.

CHAPTER 7

CONCLUSION

7.1 Contributions and impacts of the dissertation

For embedded computing platform with rapidly growing mobility requirement, low energy design has become one of the most significant issues. In this dissertation, we have studied accuracy-energy tradeoffs for DSP applications and low energy design techniques through seeking optimal tradeoffs. The entire dissertation focuses on aggressive supply voltage scaling to achieve significant reduction in energy consumption while maintaining system performance. Since the accuracy of computations is closely related to the output quality of the system, we have considered two different sources of quality degradation. In the first, the delay error due to aggressive voltage scaling is a direct source that affects the output quality of the system. In addition, the other source considered in this work is inaccurate computations to reduce or prevent the delay error. Based on the fact that the inherent error tolerance of DSP applications from the limit of human perceptual system allows relaxing 100% correctness, we have discussed ultra low energy design methodologies for DSP applications focused on efficient accuracy-energy tradeoffs.

Since aggressive voltage scaling causes delay errors under the timing constraint for nominal case, the error analysis and modeling is the fundamental step before discussing low energy design methodologies. The experimental analysis that shows the huge difference in the error rate among several delay estimation schemes clearly provides the importance of accurate delay estimation for error analysis under aggressive voltage scaling. We have presented an input sequence dependent error analysis and a model based on transition delay-based estimation. Based on the accurate error analysis, we have

found that the voltage scalability is strongly dependent on input image types. The presented experimental analysis demonstrates that images with high contrast, texture, and frequency are more error tolerant than those with opposite characteristics in terms of two aspects, the natural disparity in error tolerance due to the characteristics of the human visual system and the dependence of error rates on input image types under aggressive voltage scaling. Experimental results show that the difference in average energy savings between sharpened images and blurred images is about 17% for a given quality requirement. This analysis provides a new way to exploit optimal accuracy-energy tradeoffs for digital image processing.

We have also presented the system-level analysis of accuracy-energy tradeoffs for image processing under aggressive voltage scaling. Although aggressive voltage scaling to an energy hungry block results in significant energy savings with the acceptable degradation of output image quality, it may cause energy overhead in other components in the system. We have presented a simple system-level analysis that shows the impact of aggressive voltage scaling on the overall energy consumption of an image compression system. The erroneous operations in DCT under scaled voltages not only degrade output image quality but also reduce the compression ratio, which increases the energy consumption of memory to store the compressed image data. The proposed pixel and coefficient truncation technique effectively suppresses the increase in delay while scaling the supply voltage. Thus, the proposed technique helps to maintain high quality level by reducing the error rate at a trivial implementation cost. The experimental results demonstrate that this simple technique may result in up to 40% overall energy savings.

Finally, we have introduced an error concealment technique to achieve efficient accuracy-energy tradeoffs based on the characteristics of the image compression system. The main idea of this last work is to minimize the impact of delay errors on final output image. Instead of allowing erroneous operations that cause rapid degradation in image quality with respect to voltage scaling, we minimize the quality degradation by

substituting the erroneous output for the alternative output only at the presence of an error. For the image compression system, the proposed reduced coefficient 2-D DCT unit provides somewhat inaccurate outputs that substitute erroneous outputs to maintain high-quality level while allowing very low voltage operations when a delay error is detected in the full 2-D DCT unit. With the additional input gating and the increase in compression ratio, the proposed approach results in about 37% overall energy savings at a high-quality level (0.9 of MSSIM index) with 8.5% area increase.

In conclusion, from the fundamental studies of circuit behavior under aggressive voltage scaling to the system-level energy optimization techniques, we have explored ultra low energy design methodologies based on efficient accuracy-energy tradeoffs. The analysis and proposed techniques explained in this dissertation will have great implications for low-energy system design for error tolerant DSP applications.

7.2 Future research

In the early research, we focused on the experimental analysis to find out the relationship between voltage scalability and image characteristics. However, we have not presented a practical technique that attempts to achieve energy savings using the findings. In order to allow prediction-based adaptive parameter controls for efficient accuracy-energy tradeoffs, future research should develop methods to extract characteristic information of images at a small cost. To improve the efficiency further, this image characteristic dependent approach may also be combined to other techniques that modulate the accuracy of computations for energy savings. For example, the truncation technique presented in this dissertation may utilize the prediction information. Dependent on image characteristics, the number of bits to be truncated or voltage level can be adaptively adjusted for maximum energy savings while satisfying the quality requirement. As shown in previous work and proposed work, numerous techniques that trade off the accuracy of computations for energy savings without significant quality degradation. The

combination of these approaches may allow more fine controls over the output quality and the energy consumption of a system. However, the increased complexity for control logic and its impact on overall energy consumption will be problems to address.

APPENDIX A

SUPPLYMENTS FOR JPEG ENCODER

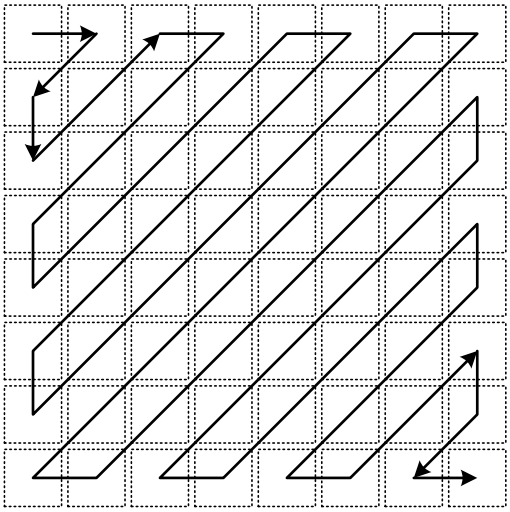


Figure A.1: Zigzag scan of DCT coefficients within an 8x8 block. Each dotted square means a DCT coefficient.

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Figure A.2: JPEG luminance quantization table.

APPENDIX A

TEST IMAGES

REFERENCES

[1] International Technology Roadmap for Semiconductors (ITRS), Available: http://www.itrs.org

[2] International Telecommunication Union (ITU), Available: http://www.itu.int

[3] Gartner, Available: http://www.gartner.com

[4] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*, Wiley, 1999

[5] A. Chandrakasan, S. Sheng, and R.W. Brodersen, "Low power digital CMOS design," *IEEE Journal of Solid State Circuits*, pp. 473-484. April 1992.

[6] A. Chandrakasan and R.W. Brodersen, "Minimizing power consumption in digital CMOS circuits," *Proceedings of the IEEE*, vol. 83, no. 4, pp. 1210-1216, Apr. 1995.

[7] N. S. Kim, T. Kgil, K. Bowman, V. De, and T. Mudge, "Total power-optimal pipelining and parallel processing under process variations in nanometer technology," *IEEE/ACM International Conference on Computer-Aided Design*, pp. 535- 540, Nov. 2005.

[8] K-S. Yeo and K. Roy, *Low-voltage, low-power VLSI subsystems*, New York: McGraw-Hill, 2005.

[9] Z. Wang and A. C. Bovik. *Modern Image Quality Assessment, in Syntheses Lectures on Image, Video and Multimedia Processing*, Morgan & Claypool Publishers, 2006.

[10] A. Chandrakasan and R. Brodersen, *Low power digital CMOS design*, Kluwer Academic Publishers, 1995.

[11] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," *IEEE Transaction on Very Large Integration Systems*, vol. 9, no. 6, pp. 813-823, Dec. 2001.

[12] R. Hegde and N. R. Shanbhag, "A voltage overscaled low-power digital filter IC," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 2, pp. 388-391, Feb. 2004.

[13] Y. Liu, T. Zhang, and J. Hu, "Low power trellis decoder with over-scaled supply voltage," *Proceedings of IEEE Workshop Signal Processing System*, pp. 205–208, 2006.

[14] N. R. Shanbhag, "Reliable and efficient system-on-chip design," *IEEE Computer*, vol.37, no.3, pp. 42- 50, Mar 2004.

[15] B. Shim; S.R. Sridhara, and N.R. Shanbhag, "Reliable low-power digital signal processing via reduced precision redundancy," *IEEE Transactions on Very Large Scale Integration Systems*, vol.12, no.5, pp.497-510, May 2004.

[16] G. V. Vartkar and N. R. Shanbhag, "Error-resilient motion estimation architecture," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 16, no. 10, Oct. 2008.

[17] R. A. Abdallah and N. R. Shanbhag, "Error-resilient low-power Viterbi decoder architecture," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 57, no. 12, Dec. 2009.

[18] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, and K. Flautner, "Razor: a low-power pipeline based on circuit-level timing speculation", *Proceedings of International Symposium on Microarchitecture*, pp. 7-18, Dec. 2003.

[19] A. Chandrakasan, W. Bowhill, and F. Fox, *Design of High-Performance Microprocessor Circuits*, IEEE Press, 2001.

[20] G. Ribes, J. Mitard, M. Denais, S. Bruyere, F. Monsieur, C .Parthasarathy, E. Vincent, and G. Ghibaudo, "Review on high-k dielectrics reliability issues," *IEEE Transactions on Device and Materials Reliability*, vol. 5, no. 1, March 2005.

[21] Y. Ye, S. Borkar and V. De, "A new technique for standby leakage reduction in high-performance circuits," *Symposium on VLSI circuits*, pp.40-41, 1998

[22] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE Journal of Solid-State Circuits*, vol.30, no.8, pp.847-854, Aug 1995.

[23] J.W. Tschanz, J.T. Kao, S.G. Narendra, R. Nair, D.A. Antoniadis, A.P. Chandrakasan, and V. De, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE Journal of Solid-State Circuits*, vol.37, no.11, pp. 1396- 1402, Nov 2002.

[24] J. Kao, S. Narendra, A. Chandrakasan, "Subthreshold leakage modeling and reduction techniques," *International Conference on Computer Aided Design*, pp. 141-148, 2002.

[25] C. Long and L. He, "Distributed sleep transistor network for power reduction," *IEEE Transactions on Very Large Scale Integration Systems*, vol.12, no.9, pp.937-946, Sep. 2004.

[26] S. Sirichotiyakul, T. Edwards, C. Oh, J. Zuo, A. Dharchoudhury, R. Panda, and D. Blaauw, "Stand-by power minimization through simultaneous threshold voltage selection and circuit sizing," *Proceedings of Design Automation Conference*, pp.436-441, 1999.

[27] J.W. Tschanz, S.G. Narendra, Y. Ye, B.A. Bloechel, S. Borkar, and V. De, "Dynamic sleep transistor and body bias for active leakage power control of microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 1, pp. 1838-1845, Nov. 2003.

[28] M. Potkonjak, M. B. Srivastava, and A. P. Chandrakasan, "Multiple constant multiplications: efficient and versatile framework and algorithms for exploring common subexpression elimination," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 2, pp. 151-165, Feb. 1996.

[29] J. Park and K. Roy, “A low complexity reconfigurable DCT architecture to trade off image quality for power consumption,” *Journal of Signal Processing Systems*, Vol. 53, No. 3, pp. 399-410, 2008.

[30] M. Potkonjak, M. B. Srivastava, and A. P. Chandrakasan, “Efficient substitution of multiple constant multiplications by shifts and additions using iterative pairwise matching,” *Proceedings of Design Automation Conference*, pp. 189-194, 1994.

[31] C.E. Leiserson and J. Saxe, “Retiming synchronous circuitry,” *Journal of Arithmetic*, vol. 6, no. 1-6, pp. 5-35, 1991.

[32] T. C. Denk and K. K.Parhi, “Two-dimensional retiming [VLSI design],” *IEEE Transactions on Very Large Scale Integration Systems*, vol.7, no.2, pp.198-211, June 1999.

[33] J. R. Jiang and R. K. Brayton, “Retiming and resynthesis: a complexity perspective,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.25, no.12, pp.2674-2686, Dec. 2006.

[34] R. Gonzalez, B.M. Gordon, and M.A. Horowitz, "Supply and threshold voltage scaling for low power CMOS," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 8, pp. 1210-1216, Aug. 1997.

[35] M. Weiser, B. Welch, A. Demers, and S. Shenker, “Scheduling for reduced CPU energy,” *Proceedings of First USENIX Symposium on Operating Systems Design and Implementation*, pp. 13–23, 1994.

[36] O. S. Unsal and I. Koren, "System-level power-aware design techniques in real-time systems," *Proceedings of the IEEE*, vol. 91, no.7, pp. 1055- 1069, July 2003.

[37] G. Semeraro, G. Magklis, R. Balasubramonian, D.H. Albonesi, S. Dwarkadas, and M.L Scott, “Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling,” *International Symposium on High-Performance Computer Architecture*, pp. 29- 40, 2-6 Feb. 2002.

[38] K. Choi, K. Dantu, W.-C. Cheng, and M. Pedram, “Frame-based dynamic voltage and frequency scaling for a MPEG decoder,” *International Conference on Computer Aided Design*, pp. 732- 737, Nov. 2002.

[39] D. Rakhmatov and S. Vrudhula, “Energy management for battery-powered embedded systems,” *ACM Transactions on Embedded Computing Systems*, vol. 2, no. 3, Aug. 2003.

[40] K. Choi, R. Soma, and M. Pedram, “Dynamic voltage and frequency scaling based on workload decomposition,” *International Symposium on Low Power Electronics and Design*, pp.174-179, 11-11 Aug. 2004.

[41] K. A. Bowman, S. G. Duvall, and J. D. Meindl, “Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration,” *IEEE Journal of Solid-State Circuits,* vol. 37, no. 2, pp.183-190, Feb 2002.

[42] S.G. Narendra and A. Chandrakasan, *Leakage in Nanometer CMOS Technologies*. Springer Publications, 2006.

[43] A. Asenov, “Random dopant induced threshold voltage lowering and fluctuations in sub-0.1 μm MOSFET's: A 3-D “atomistic” simulation study," *IEEE Transactions on Electron Devices,* vol.45, no.12, pp.2505-2513, Dec 1998.

[44] S. S. Sapatnekar, “Overcoming variations in nanometer-scale technologies,” *IEEE Transactions on Emerging and Selected Topics in circuits and Systems*, vol. 1, no. 1, March 2011.

[45] J.C. Ku, and Y. Ismail, "On the scaling of temperature-dependent effects," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol.26, no.10, pp.1882-1888, Oct. 2007.

[46] Joint Photograph Experts Group (JPEG). http://www.jpeg.org.

[47] K. R. Rao and P. Yip, *Discrete cosine transform: Algorithms, advantages, applications*. Boston: Academic, 1990.

[48] A. Ligtenberg and M. Vetterli, “A discrete Fourier-cosine transform chip”, *IEEE Journal on Selected Areas in Communications SAC-4*, 49-61, Jan. 1986.

[49] W.-H. Chen, C. H. Smith, and S. Fralick, “A fast computational algorithm for the discrete cosine transform", *IEEE Trans. on Communications*, vol. 25, no. 9, 1004–1009, Sep. 1977.

[50] S.C. Chan and K. L. Ho, “A new two-dimensional fast cosine transform,” *IEEE Transaction on Signal Processing*, vol. 39, pp. 481–485, 1991.

[51] N. I. Cho and S. U. Lee, “Fast algorithm and implementation of 2-D discrete cosine transform,” *IEEE Transaction on Circuits Systems*, vol. 38, pp.297–305, 1991.

[52] Y. Arai, T. Agui, and M. Nakajima, “A fast DCT-SQ scheme for images,” *Transactions of IEICE*, vol. E71, pp. 1095-1097, 1988.

[53] W. B. Pennebaker and J. L. Mitchell, *JPEG: still image data compression standard*, Kluwer Academic Publishers, 1992.

[54] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli., “Image quality assessment: from error measurement to structural similiarity," *IEEE Transactions on Image Processing*, vol. 13, no. 6, pp. 600-612, 2004.

[55] H. R. Sheikh, A. C. Bovik, and G. de Veciana, “An information fidelity criterion for image quality assessment using natural scene statistics,” *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2117–2128, 2005.

[56] A. Shnayderman, A. Gusev, and A. M. Eskicioglu, “An SVD-based grayscale image quality measure for local and global assessment,” *IEEE Transactions of Image Processing*, vol. 15, no. 2, pp. 422–429, 2006.

[57] D. M. Chandler and S. S. Hemami, “VSNR: a wavelet-based visual signal-to-noise ratio for natural images,” *IEEE Transactions on Image Processing*, vol.16, no.9, pp.2284-2298, Sep. 2007.

[58] M. A. Breuer, S. K. Gupta, and T.M. Mak, "Defect and error tolerance in the presence of massive numbers of defects," *IEEE Design and Test of Computers*, vol. 21, no. 3, pp. 216-227, May 2004.

[59] M. A. Breuer, "Multi-media applications and imprecise computation," *Proceedings of Euromicro conference on Digital System Design*, pp.2-7, 2005

[60] S. H. Nawab, A. V. Oppenheim, A. P. Chandrakasan, J. M.Winograd, and J. T. Ludwig, "Approximate signal processing," *Journal of VLSI Signal Processing*, vol. 15, no. ½ , pp. 177-200, Jan. 1997.

[61] S. Hua, G. Qu, and S. S. Bhattacharyya, "Energy reduction techniques for multimedia applications with tolerance to deadline misses," *Proceedings of Design Automation Conference*, pp. 131-136, 2003.

[62] F.W. Campbell and J.G. Robson, "Application of Fourier analysis to the visibility of gratings," Journal of Physiology, vol. 197, no. 3, pp. 551-561, 1968.

[63] Y. Q. Shi and H. Sun, *Image and Video Compression for Multimedia Engineering*, CRC Press, 2000.

[64] F. Fang, T. Chen, and R. A. Rutenbar, "Lightweight floating-point arithmetic: case study of inverse discrete cosine transform," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 1, Jan. 2002.

[65] I-M. Pao and M-T. Sun, "Modeling DCT coefficients for fast video encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, June 1999.

[66] N. J. August and D. S. Ha, "Low power design of DCT and IDCT for low bit rate video codecs," *IEEE Transactions on Multimedia*, vol. 6, no. 3, pp. 414- 422, June 2004.

[67] Z. L. He, C.Y. Tsui, K. K. Chan, and M.L. Liou, "Low-power VLSI design for motion estimation using adaptive pixel truncation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 5, pp. 669-678, 2000.

[68] A. Bahari, T. Arslan, and A. T. Erdogan, "Low-power H.264 video compression architectures for mobile communication," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 9, June 2009.

[69] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos, "Data driven signal processing: an approach for energy efficient computing," *International Symposium on Low Power Electronics and Design*, pp.347-352, Aug 1996.

[70] K. V. Palem, "Energy aware computing through probabilistic switching: a study of limits," IEEE Transactions on Computers, vol. 54, no. 9, 2005.

[71] L. N.Chakrapani, B.E.S. Akgul, S. Cheemalavagu, P. Korkmaz, K.V. Palem, and B. Seshasayee, "Ultra-efficient (embedded) SOC architectures based on probabilistic CMOS (PCMOS) technology," *Proceedings of Design, Automation and Test in Europe*, pp.1-6, Mar. 2006.

[72] J. George, B. Marr, B. E. S. Akgul, and K.V. Palem, "Probabilistic arithmetic and energy efficient embedded signal processing," *International Conference on Compilers, Architecture, and Sysnthesis for Embedded Systems*, pp. 158-168, 2006.

[73] L. N. B. Charkrapani, K. K. Muntimadugu, A. Lingamneni, J. George, and K. V. Palem, "Highly energy and performance efficient embedded computing through approximately correct arithmetic," *International Conference on Compilers, Architecture and Synthesis for Embedded Systems*, pp. 187-196, 2008.

[74] F. Kurdahi, A. Eltawil, A.K. Djahromi, M. Makhzan, and S.Cheng, "Error-aware design," *Euromicro Conference on Digital System Design*, pp. 8-15, 2007.

[75] D. Blaauw, S. Kalaiselvan, K. Lai, M. Wei-Hsiang, S. Pant, C. Tokunaga, S. Das, and D. Bull, "Razor II: in situ error detection and correction for PVT and SER tolerance," *IEEE International Solid-State Circuits Conference*, pp. 400-622, Feb. 2008.

[76] H. Fuketa, M. Hashimoto, Y. Mitsuyama, and T. Onoye, "Trade-off analysis between timing error rate and power dissipation for adaptive speed control with timing error prediction," *Proceedings of Asia and South Pacific Design Automation Conference*, pp. 266-271, Jan. 2009.

[77] S. Ghosh, S. Bhunia, and K. Roy, "CRISTA: a new paradigm for low-power variation-tolerant, and adaptive circuit synthesis using critical path isolation," *IEEE Transaction on Very Large Integration System*, vol. 26, no. 11, pp. 1947-1956, Nov. 2007.

[78] S. Ghosh, D. Mohapatra, G. Karakonstantis, and K. Roy, "Voltage scalable high-speed robust hybrid arithmetic units using adaptive clocking," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 18, no. 9, Sep. 2010.

[79] N. Banerjee, G. Karakonstantis, J. H. Choi, C Chacrabarti and K. Roy, "Design methodology for low power dissipation and parametric robustness through output quality modulation: Application to Color Interpolation Filtering," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 8, Aug. 2009.

[80] V. Joshi, D. Blaauw, and D. Sylvester, "Soft-edge flip-flops for improved timing yield: design and optimization," *International Conference on Computer-Aided Design*, pp.667-673, Nov. 2007.

[81] K. Chae, S. Mukhopadhyay, C. H. Lee, and J. Laskar, "A dynamic timing control technique utilizing time borrowing and clock stretching," *IEEE Custom Integrated Circuit Conference*, 2010.

[82] Predictive Technology Model (PTM), Available: http://www.eas.asu.edu/~ptm.

[83] Synopsys, Available: http://www.synopsis.com.

[84] C. T. Gary, W. Liu, R. K. Cavin, and H. Hsieh, "Circuit delay calculation considering data dependent delays," *Integration, the VLSI Journal*, pp.1-23, 1994.

[85] S. Sun, D. H. C. Du, and H. Chen, "Efficient timing analysis for CMOS circuits considering data dependent delays," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.7, no. 6, pp. 546-552, Jun. 1998.

[86] M. D. Ercegovac and T. Lang, *Digital Arithmetic*, Morgan Kaufmann, 2003.

[87] M. Cho, J. Schlessman, W. Wolf, and S. Mukhopadhyay, "Accuracy-aware SRAM: a reconfigurable low power SRAM architecture for mobile multimedia applications," *Proceedings of Asia and South Pacific Design Automation Conference*, pp. 823-828, 2009.

[88] K. He, A. Gerstlauer, and M. Orshansky, "Controlled timing-error acceptance for low energy IDCT Design," *Design, Automation & Test in Europe Conference & Exhibition*, pp.1, March 2011.

[89] M. Kovac and N. Ranganathan, "JAGUAR: a fully pipelined VLSI architecture for JPEG image compression standard," *Proceedings of the IEEE*, vol. 83, no. 2, pp. 247-258, 1995.

[90] L. V. Agostini, S. Bampi, and I. S. Silva, "Pipelined fast 2D DCT architecture for JPEG image compression," *Symposium on Integrated Circuits and Systems Design*, pp. 226-231, 2001.

[91] MediaBench, Available: http://euler.slu.edu/~fritts/mediabench.

[92] http://www.imagecompression.info/test_images/

[93] http://www.imageprocessing.com.

[94] http://sipi.usc.edu/database/index.html.

[95] J. M. Rabaey and M. Pedram, *Low power design methodologies*, Norwell, MA; Kluwer academic publishers, 1996.

[96] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, NJ; Rearson/Prentice Hall, 2008.

[97] Micron Technology, Inc. TN-46.12. http://www.micron.com/support/part_info/powercalc.

[98] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi, *Low power methodology manual: for system-on-chip design*, Springer, 2007.

[99] FreePDK, Available: http://www.eda.ncsu.edu/wiki/FreePDK.